Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Dejan Kljajić

Varnostni vidik pri implementaciji pametne hiše

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: dr. Branko Šter

Ljubljana, 2016

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Preučite načine implemetacije pametne hiše. Opišite morebitne grožnje, ki pretijo hišam s takim sistemom, in razložite, kako bi jih zmanjšali ali se jim izognili. Na podlagi preučene snovi realizirajte sistem s centralno enoto, aplikacijo, mikrokrmilnikom in senzorji. Posebno pozornost namenite zaščiti omrežja in varnosti posameznih komponent sistema.

Hvala družini in vsem ostalim, ki so me podpirali v času študja. Zahvaljujem se tudi dr. Branku Šteru za mentorstvo.

Bratu Aleksu.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
API	Application Programming Interface	Aplikacijski programski vmesnik
CSS	Cascading Style Sheets	Kaskadne stilske podloge
DDoS	Distributed Denial of Service	Porazdeljena zavrnitev storitve
DHCP	Dynamic Host Configuration Protocol	Omrežni protokol za dinamično nastavitev gostitelja
DMZ	Demilitarized Zone	Demilitarizirano območje
DNS	Domain Name System	Sistem domenskih imen
DoS	Denial of Service	Zavrnitev storitve
HTML	Hyper Text Markup Language	Jezik za označevanje nadbesedila
HTTP	Hypertext Transfer Protocol	Komunikacijski protokol za prenos informacij na spletu
HTTPS	HTTP Secure	Varna verzija HTTP
ICMP	Internet Control Message Protocol	Protokol za pošiljanje nadzornih sporočil
IoT	Internet of Things	Internet stvari
IP	Internet Protocol	Internetni protokol
ISP	Internet Service Provider	Internetni ponudnik
IPS	Intrusion Prevention System	Sistem za preprečevanje vdorov

kratica	angleško	slovensko
LAN	Local Area Network	Lokalno omrežje
MAC	Media Access Control	Nadzor dostopa do medija
MVC	Model-View-Controller	Model-pogled-nadzornik
NAS	Network-attached storage	Naprava na omrežju za shranjevanje datotek
QoS	Quality of service	Zagotavljanje kakovosti prenosa
RADIUS	Remote Authentication Dial-In User Service	Omrežni protokol za avtantikacijo, avtorizacijo in obračunavanje
SSH	Secure Shell	Protokol za varno upravljanje naprav v omrežju
SSID	Service Set Identification	Identifikator v brezžičnem omrežju
SSL	Secure Sockets Layer	Protokol za varno komunikacijo
TCP	Transmission Control Protocol	Protokol za nadzor prenosa
TFTP	Trivial File Transfer Protocol	Preprost protokol za prenos datotek
UDP	User Datagram Protocol	Nepovezovalni protokol za prenašanje paketov
UPS	Uninterruptible power supply	Brezprekinitveno napajanje
UTM	Unified Threat Management	Večnamenska požarna pregrada
UTP	Unshielded Twisted Pair	Neoklopljena sukana parica
VLAN	Virtual LAN	Navidezno lokalno omrežje
VPN	Virtual Private Network	Navidezno zasebno omrežje
WEP	Wired Equivalent Privacy	Varnostni algoritem v brezžičnih omrežjih

Povzetek

Naslov: Varnostni vidik pri implementaciji pametne hiše

Avtor: Dejan Kljajić

Z razmahom pametnih naprav in sistemov za nadzor hiše se je uveljavil izraz pametna hiša, ki označuje dom z medsebojno povezanimi napravami. Veliko naprav pomeni veliko tehnologij različnih proizvajalcev, za katere sta značilna čim večje število funkcij in nizka varnost. V diplomski nalogi so predstavljene grožnje, ki omenjenim sistemom in napravam pretijo, in njihove morebitne rešitve. Cilj naloge je preučiti grožnje in dobre prakse ter jih uporabiti kot vodilo pri izgradnji sistema za pametno hišo. Implementacija je razdeljena na tri dele: omrežni del, spletno aplikacijo na centralni enoti in mikrokrmilnike kot končne naprave. Ugotovili smo, da so grožnje realne, in ker so tehnologije za njihovo preprečitev na voljo, smo jih tudi implementirali. Naloga zaradi podrobnejših opisov lahko služi kot osnova za izdelavo podobnih sistemov.

Ključne besede: pametna hiša, IoT, pametne naprave, varnost, OpenWrt, Raspberry Pi, aplikacija, Arduino.

Abstract

Title: Safety Aspect in the Implementation of Smart Homes

Author: Dejan Kljajić

With the rise in smart devices and home control systems, the term Smart House, which denotes a home incorporating interconnected devices, became significantly more frequent. A large number of devices means a large number of technologies developed by various manufacturers and characterised by numerous functions and low safety. Besides explaining the threats to the above-mentioned systems and devices, this thesis also presents their possible solutions. The purpose of this work is to thoroughly examine the threats and good practices, and then use them as guidelines for building a Smart House system. The implementation is divided into three parts: network, web application on the main unit, and microcontrollers as end devices. We discovered that the threats are real, and since the technologies for their prevention are available, we implemented them. Because of its detailed descriptions, the thesis can serve as a basis for building similar systems.

Keywords: Smart home, IoT, smart devices, security, OpenWrt, Raspberry Pi, application, Arduino.

Poglavje 1

Uvod

Pametna hiša označuje sistem povezanih naprav, ki uporabnikom omogoča prijetnejše bivanje, večjo varnost in nadzor nad hišo. Osnovne funkcije, ki jih pričakujemo, so: upravljanje s svetili, nadzor porabe, nadzor naprav, samodejno upravljanje z ogrevanjem ipd.

Implementacije hišne avtomatizacije so v zadnjih petih letih močno v porastu [20]. Z napredkom tehnologije in nižjimi cenami so sistemi postali bolj dostopni množicam. V grobem lahko ločimo tri načine implementacije. Prva možnost je izbira enega proizvajalca z njegovim sistemom, kar je običajno najdražje in navadno zahteva načrtovanje sistema že pred izgradnjo objekta. Druga možnost je nakup različnih namenskih naprav, ki se povežejo na domač usmerjevalnik in dostopajo do spletnih storitev, kamor zapisujejo podatke, do katerih mi dostopamo z aplikacijo. Tretja možnost je, da zgradimo svoj sistem z različnimi senzorji in napravami, za postavitev katerega je treba imeti dovolj znanja in časa. Pri vseh možnostih pa je ključna komponenta, ki se jo zaradi težavnosti in časovne zahtevnosti večkrat prezre, varnost.

Uporabniki, ki naprave namestijo v svoje domove, običajno niso seznanjeni z varnostnimi tveganji oz. se jih ne zavedajo, ki jih te naprave prinašajo. S porastom pametnih hiš se povečuje tudi tveganje za vdor v sisteme. Napadalci lahko pridobijo nadzor nad hišo, ne le to, pridobijo tudi dostop do občutljivih informacij, kot so navade stanovalcev. Krivce je pri takih vdorih

težko najti. Po eni strani so to neprevidni in nezavedni uporabniki, po drugi strani pa sistem sam in tisti, ki ga upravljajo. V izogib podobnim situacijam je trenutno na voljo zelo malo literature, ki bi na enem mestu opisala zaščito takih implementacij. Na internetu je moč dobiti dve študiji Agencije Evropske unije za varnost omrežij in informacij (krajše ENISA): prvo z naslovom Threat Landscape for Smart Home and Media Convergence [7] in drugo z naslovom Security and Resilience of Smart Home Environments [6]. V prvi študiji so opisane grožnje, ki pretijo pametnim hišam, v drugi študiji pa so opisana dobra praksa in priporočila pri vpeljavi naprav IoT (ang. Internet of Things; sl. internet stvari) v domove.

1.1 Cilj diplomske naloge

Cilj diplomske naloge je preučiti varnostna tveganja, ki jih taki sistemi prinašajo s seboj, in zanje poiskati primerne rešitve, obenem pa delujočo rešitev tudi realizirati. Cilj bomo dosegli z razdelitvijo dela na več segmentov. Najpomembnejši del naloge bo zajemal implementacijo domačega omrežja s poudarkom na varnosti. Sledil bo opis aplikacije, primerne za nadzor in dostop do sistema, in njene namestitve. Za centralno enoto, na kateri bo gostovala aplikacija, bo uporabljen računalnik Raspberry Pi, ki bo komuniciral z mikrokrmilnikom Arduino. Na Arduinu bodo priključeni senzor in naprave, ki bodo simulirale komponente v pametni hiši.

V prvem delu bo predstavljen postopek nastavitve sistema OpenWrt na usmerjevalnik. Za upravljanje z uporabniki na brezžičnem omrežju bo skrbel RADIUS (ang. Remote Authentication Dial-In User Service; sl. Omrežni protokol za avtentikacijo, avtorizacijo in obračunavanje), omrežje in požarni zid pa bosta nastavljena podobno kot v podjetjih, kjer so varnostne politike izredno stroge.

Aplikacija bo napisana v jeziku PHP [11]. Zaradi same narave implementacije bosta varnost in način interakcije imela večjo težo od ostalih vidikov pri ustvarjanju aplikacij. Centralna enota bo realizirana na Raspberry Piju,

računalniku velikosti kreditne kartice. Ta bo delovala kot posrednik med Arduinom, ki bo podatke pridobil iz senzorjev, in aplikacijo, ki bo te podatke prikazala, oz. obratno, za nastavljanje parametrov napravam in senzorjem.

V zadnjem poglavju implementacije bodo predstavljeni pogosti načini realizacije brezžične komunikacije med napravami Arduino, varnost teh načinov in njihova primernost glede na podatke, ki se pošiljajo.

Sklepni del bo vseboval celoten pregled in zaključke, ki bodo temeljili na preučeni snovi in realizirani rešitvi.

Poglavje 2

Pregled področja

Glede na podatke iz storitve Google Trends [10] je od sredine julija leta 2013 v porastu iskanje besede IoT. IoT je koncept med seboj povezanih naprav, ki zajemajo in procesirajo podatke, ki se uporabljajo v določenem kontekstu. Primer take naprave je pametna sijalka, ki se poveže na domačo dostopno točko, podatke pošilja in sprejema iz oblačne storitve, kot uporabniki pa imamo možnost nadzora naprave preko aplikacije.

Skupek takih naprav je mogoče uporabiti v t.i. pametnih hišah, a je pri njihovi uporabi treba paziti, saj nepravilno izdelane ali nameščene naprave prinašajo s seboj številne nevarnosti [7]. Kljub temu, da so bile naprave sprva mišljene za izboljšanje življenja, lahko škodujejo uporabnikom, v skrajnih primerih tudi ogrozijo življenje.

2.1 Definicija pametne hiše in njena funkcija

Pojem pametne hiše še ni bil natančno opredeljen. Eno izmed definicij pametne hiše na primer podaja Vitez, ki pravi: "inteligentna hiša je tista, ki ima vgrajeno aktivno inteligenco, katera je namenoma vgrajena v objekt in je sposobna zaznati parametre iz okolja in se nanje odzivati skladno z vnaprej določenim algoritmom" [23]. Sam bi pametno hišo opredelil kot hišo z vgrajenim centraliziranim sistemom, sestavljenim iz ene glavne enote in več

različnih namenskih naprav, ki uporabnikom omogočajo nadzor nad hišo ter varno in udobno bivanje. Dodal bi še, da mora biti tak sistem popolnoma varen, zgrajen skladno s standardi na vseh področjih, robusten in fleksibilen oz. prilagodljiv za spremembe. Funkcionalnosti pametne hiše lahko v grobem delimo na tri dele [12]:

- upravljanje z energijo, vodo, ogrevanjem, prezračevanjem;
- varnost hiše;
- izboljšanje kakovosti življenja.

Slika 2.1: Funkcije pametne hiše.

Vse naprave in senzorji v sistemu pošiljajo informacije centralni enoti, ki na podlagi sprejetih parametrov ustrezno reagira. Tako je lahko senzor, ki je nameščen zunaj hiše in primarno uporabljen kot nadzor za porabo električne energije, uporabljen tudi kot eden od parametrov za varovanje hiše.

Poleg centralne enote sta pomembna še dva dela sistema. Prvi je aplikacija, ki uporabniku omogoča dostop do podatkov, spreminjanje parametrov in splošen nadzor nad hišo. Poleg vseh naštetih funkcionalnosti je treba upoštevati kakovost izdelave. Ali so pri razvoju upoštevali dobre prakse,

na kakšen način se v aplikacijo vpišemo, kako so hranjena gesla. Podobna vprašanja si bomo zastavili v kasnejšem poglavju, na katera bomo pri implementaciji praktično odgovorili z realizacijo. Druga pomembna komponenta sistema je omrežje. Današnji domači usmerjevalniki imajo veliko varnostnih funkcij, ki so že omogočene ob prvem vklopu, vendar te ne zadostujejo za omrežje, v katerem neznanci (gostiteljsko omrežje) nimajo kaj početi. Na trgu obstajajo celotne naprave, t.i. požarni zidovi naslednje generacije (ang. Unified Threat Management; krajše UTM). Ti poleg možnosti usmerjevalnika in stikala vsebujejo še VPN (ang. Virtual Private Network; sl. navidezno zasebno omrežjee), požarni zid, IPS (ang. Intrusion Prevention System; sl. sistem za preprečevanje vdorov), protivirusno zaščito privzetega prehoda, filtra za neželeno pošto, omejitve dostopa do spletnih strani, QoS (ang. Quality of service; sl. zagotavljanje kakovosti prenosa), preprečitev izgube podatkov idr. [32]. Naprave UTM za podjetja niso novost, so se pa zaradi nižanja cen pojavile tudi v domači uporabi.

2.2 Možnosti implementacije pametne hiše

Potrošniki imamo trenutno tri možnosti za realizacijo pametne hiše (današnje stanje). Vsaka od njih ima svoje prednosti in slabosti, odločimo se glede na našo trenutno situacijo in finančne zmožnosti.

2.2.1 Nakup celotnega sistema priznane znamke

To je najlažja in najvarnejša možnost. Izvedljiva je ne glede na stanje hiše, priporočljivo pa je implementiranje takega sistema ob renoviranju ali novi gradnji. Takrat namreč projektanti najlažje določijo, kje bodo posamezne naprave in senzorji. Izbrano podjetje postavi sistem, predstavi uporabo aplikacije in vam v primeru težav nudi podporo. Pri taki implementaciji je morda bolje imeti napravo, kot je UTM, ki se nastavi in deluje dalj časa. Taka rešitev je najprimernejša za večino ljudi. Težave takega sistema so, da smo vezani na enega ponudnika oz. znamko, omejeni smo na naprave, ki

jih v takem sistemu nudijo (heterogeni sistem običajno ni mogoč), zaupati moramo skrbniku sistema (podjetju oz. zastopniku) in navsezadnje je treba upoštevati tudi finančni vidik.

2.2.2 Nakup več različnih naprav različnih proizvajalcev

Z nakupom naprav različnih proizvajalcev ustvarimo heterogen sistem, ki ga lahko postavimo v že urejeni hiši. O heterogenem sistemu govorimo, ko en proizvajalec ne nudi vseh možnih senzorjev in naprav, kot bi jih dobili pri sistemu, omenjenem v prejšnjem poglavju. Take naprave se običajno povežejo na domače omrežje in nato v oblačno storitev, do katere dostopamo z namensko aplikacijo za mobilne naprave ali s spletnim vmesnikom.

Kupovanje različnih naprav za različne namene je vsekakor praktično, saj kupimo le tiste rešitve, ki jih potrebujemo. Po drugi strani imamo tako lahko pet naprav štirih različnih proizvajalcev, do katerih dostopamo preko štirih različnih aplikacij. V tem primeru bi potrebovali centralni sistem ali komunikacijo med temi napravami, kar pa običajno ni mogoče. Vsak proizvajalec ima svoje implementacije in rešitve, običajno ne uporablja standardov, zato so kupci takih naprav obsojeni na uporabo več aplikacij [15]. Če je podatke iz oblaka možno pridobiti, lahko uporabimo centralno storitev, kot na primer IFTTT [26]. Z vidika varnosti takega sistema je pozitivno dejstvo, da varnost ne sloni na enem proizvajalcu, ampak je porazdeljena (če se ugotovi varnostna luknja pri enem, še ne pomeni, da jo imajo vsi).

Naslednja težava je dostop naprav do lokalnega omrežja. Omrežje je treba zaščititi, kar je od povprečnega uporabnika, ki nima znanja o računalniških omrežjih, popolnoma nemogoče pričakovati. Informacije se tako morda prenašajo po nezaščitenih povezavah; če so te informacije občutljive narave (npr. video prenos IP (ang. Internet Protocol; sl. internetni protokol) kamere), pa je težava še toliko večja.

Največja težava teh naprav je, da so razvite tako, da imajo čim več funkcij, z manjšim razmislekom o varnosti.

Če se odločimo za tako rešitev, je treba zagotoviti varnost domačega omrežja (več o tem v nadaljevanju), imeti dobra gesla in jih redno menjavati ter redno posodabljati naprave.

2.2.3 Lastna rešitev

Tretja možnost je, da stvari vzamemo v svoje roke in sami implementiramo celoten sistem. Hiša (dom) je lahko še v fazi načrtovanja ali že obstoječa, saj sami izberemo, kdaj in kako bomo implementacijo izvedli. Za postavitev celotnega sistema moramo sami izbrati komponente, treba pa je imeti tudi dovolj volje, časa in znanja z različnih področjih.

Možnosti za napake oz. slabo implementacijo v takem primeru so izredno visoke. Izognemo se jim lahko tako, da z vsakega področja upoštevamo najboljše prakse in uporabljamo najnovejše različice programske opreme.

Mogoče je tudi sestaviti sistem z nekaj pametnimi napravami, ki so del večjega sistema in s katerimi pridobivamo ali nastavljamo določene parametre. Ob nakupu moramo veliko pozornost nameniti varnostim omejitvam naprave in funkcijami, ki jih ponuja. Držimo se pravila, da ima varnost prednost pred množico funkcij [7].

Postavljanje takega sistema vsekakor ni enostavno, zato bodo v nadaljevanju opisane nevarnosti te implementacije in realizacija, ki lahko služi kot začetek oz. dobra osnova vzpostavitev sistema pametne hiše.

2.2.4 Varnost

V prejšnjih podpoglavjih se je velikokrat ponovila beseda varnost. Pametnih hiš je iz leta v leto več, postale so dostopne zaradi pocenitve tehnologije [14]. Trg deluje na način, da zadovolji kupca. Če kupec želi več funkcij, bodo izdelovalci naprav posvetili več sredstev v slednjo domeno [7]. Tako zmanjka finančnih sredstev za razvoj varnih naprav, saj končnega uporabnika ne zanima način, kako so stvari implementirane, ampak da za vložen denar dobi čim več funkcij. Podajmo primer. Brezžična IP kamera, ki je povezana na

domače omrežje in shranjuje podatke na NAS. Če je na kameri implementiran le protokol WEP (ang. Wired Equivalent Privacy; sl. varnostni algoritem v brezžičnih omrežjih) za povezavo v omrežje, to predstavlja večjo varnostno luknjo v sistemu, ki jo je mogoče izkoristiti za vdor v celoten sistem.

Pametna hiša je stičišče povezane informacijske tehnologije in fizičnega sveta, kar predstavlja do sedaj še nepoznane grožnje in ranljivosti [7]. Senzorji in naprave v sistemu ustvarjajo velike količine občutljivih podatkov o aktivnostih in stanju prebivalcev ter same hiše.

ENISA je izdala dve študiji o pametnih hišah, ki bosta v veliko oporo pri izdelavi naloge [6, 7]. Obe pokrivata podobno področje, ena z večjim poudarkom na varnosti, druga poudarja možne grožnje, obe pa opisujeta dobre prakse in napotke, ki se jih bomo v nadaljevanju držali.

2.3 Grožnje

Pametno hišo ogrožajo različni tako fizični kot informacijski dejavniki. Naslednja poglavja vsebujejo povzetek groženj skupin iz študije Threat Landscape for Smart Home and Media Convergence [7]. Za vsako grožnjo bo predstavljen primer zaščite s konkretno rešitvijo. Ne glede na to, ali želimo realizirati pametno hišo sami, z določenimi napravami, kombinacijo obojega ali kupiti celoten sistem, je pomembno, da se groženj in nevarnosti zavedamo, saj jih tako lahko zmanjšamo.

2.3.1 Fizični napadi

Naprave v sklopu pametne hiše lahko utrpijo poškodbe ali krajo, kar privede do prekinjene povezave med napravo in sistemom. Pod večjo nevarnostjo so naprave in senzorji, nameščeni v okolici hiše. Fizična zaščita naprav je pomembna, saj mnogi proizvajalci predpostavljajo, da imajo le uporabniki fizičen dostop do naprave. Fizičen dostop do naprave običajno omogoča tudi nalaganje programske opreme, dodajanje strojne opreme, spreminjaje nastavitev in pridobivanje podatkov z naprave.

Naša rešitev bi bila naslednja. Če naprava iz kateregakoli razloga ni dosegljiva ali pošilja napačne podatke, jo, odvisno od naprave, sistem lahko sam ponastavi na privzete nastavitve in odstrani iz sistema.

2.3.2 Nenamerna škoda

Nenamerna škoda lahko nastane v naslednjih primerih.

- Zaradi slabo implementiranih varnostnih nastavitev oz. slabega načrtovanja pride do uhajanja občutljivih informacij.
- Napake pri nastavitvi parametrov (še posebej nevarno pri sistemih z učenjem uporabnikovih navad in sistemih z glasovno prepoznavo ukazov).

- Uporaba parametrov, pridobljenih iz slabo delujoče ali ogrožene naprave.

V izogib nenamerni škodi uporabljamo obstoječe in preverjene varnostne protokole, spreminjanje nastavitev parametrov v vmesniku omejimo na vnaprej določene vrednosti (določimo meje).

2.3.3 Naravne nesreče in izpadi

V primeru naravnih nesreč ali izpadov mora sistem uporabnika obvestiti o stanju, sistem ne sme odpovedati ob izpadu električnega napajanja, ne dopustimo izgube povezave z internetom in podatkov .

Rešitev lahko najdemo v aplikaciji, ki skrbi za varnostne kopije podatkov, naloženih na oblačno storitev. Ob nesreči aplikacija glede na parametre, pridobljene iz naprav, obvesti stanovalce o stanju (požar, poplave, potres itd.). Centralno enoto namestimo v varen prostor (ognjevaren, varen pred poplavo ipd.) in jo priklopimo na brezprekinitveni sistem napajanja (UPS; ang. Uninterruptible power supply) ter jo povežemo z internetom po redundantni povezavi prek mobilnega omrežja (bazne postaje so opremljene z UPS-ji in agregati [19]), ki se aktivira ob prekinjeni povezavi do ISP-ja (ang. Internet Service Provider; sl. internetni ponudnik).

2.3.4 Izgube ali odtujitve informacij

Kot smo omenili, pametne hiše generirajo in hranijo velike količine občutljivih podatkov. Napravam lahko uhajanje podatkov pripišemo iz različnih razlogov:

- so brez varnostne programske opreme in šifriranih povezav;
- slaba implementacija oz. zasnova;
- dodatni stroški pri zasnovi;
- premajhna procesorska moč ali napajanje.

Grozita nam lahko izguba podatkov iz domačega NAS-a (ang. Network-Attached Storage; sl. naprava na omrežju za shranjevanje datotek) in izguba podatkov, če naš ponudnik oblačne storitve preneha z delovanjem.

Možne so naslednje rešitve:

- uporabimo obstoječe in preverjene varnostne rešitve;
- sistem razvijemo s poudarkom na varnosti, kljub temu, da zahteva več časa in finančnih sredstev;
- prilagodimo namestitev naprave, da bo komunikacija s centralno enoto varna (uporaba žične povezave namesto brezžične);
- podatke hranimo na več lokacijah (lokalno in na oblačni storitvi).

2.3.5 Odpovedi in slabo delovanje

Večjo težavo kot odpoved ali slabo delovanje naprav ali senzorjev predstavlja odpoved delovanja centralne naprave ali ukinitev oblačne storitve.

Rešitev bi bila imeti vse lokalno (le varnostne kopije v oblaku). Odpoved glavne naprave preprečimo s t.i. gručo (cluster); to je priporočljiv način, ki ne dopušča izpada sistema. Imamo dve enaki napravi, ki sta med seboj povezani, ena ves čas pošilja drugi trenutno stanje in vse parametre. Ko naprava preneha delovati, vse naloge prevzame druga naprava, pri tem pa druge naprave ne občutijo nobene spremembe. Aplikacija o takem dogodku obvesti odgovorno osebo za vzdrževanje sistema.

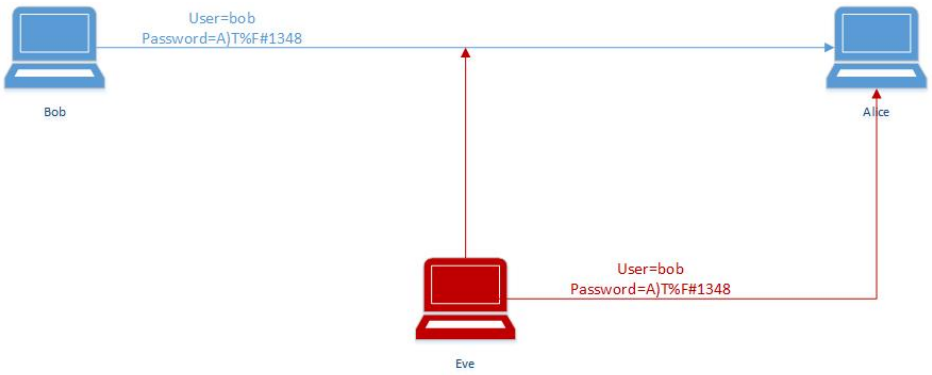
2.3.6 Prisluškovanje, prestrezanje, ugrabitve

V pametni hiši je običajno uporabljenih več načinov komunikacije med napravami. Vsak ima svoje prednosti in slabosti, ki jih moramo oceniti glede na varnost. Wi-Fi, Z-wave, ZigBee in Bluetooth so trenutno najbolj uporabljeni načini brezžične komunikacije, ki je v primerjavi z žično komunikacijo

občutljivejša na prisluškovanje, prestrezanje in vdiranje. Ker je zrak skupen medij, lahko naprave z brezžično komunikacijo (npr. senzorje, kamere) z motilcem signala onesposobimo.

Ponovitev sporočil

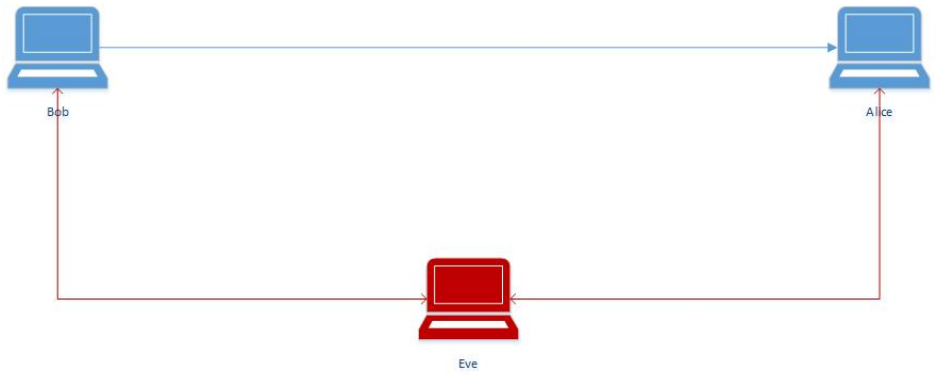
Napad se izvede s ponovnim pošiljanjem prestreznih paketov, tako naprava ponovi predhodno akcijo. ZigBee je primer protokola, ki ima minimalno zaščito pred takimi napadi [3].



Slika 2.2: Slikovna ponazoritev napada s ponovitvijo sporočil.

Napad s posrednikom

Napad s posrednikom (ang. man-in-the-middle attack) se izvede tako, da se napadalec postavi med dve napravi, z vsako vzpostavi neodvisno povezavo, kar omogoči prisluškovanje [29]. Če je to mogoče izvesti, pomeni, da ni šifrirane povezave in ni avtentikacije končnih naprav. S pridobitvijo nadzora nad napravo ima napadalec možnost prirediti podatke v dnevniku (log) in s tem prikriti svoje sledi.



Slika 2.3: Slikovna ponazoritev napada s posrednikom.

Pri vseh varnostno kritičnih napravah (kamere in ostale naprave za varnost hiše) je priporočljiva je uporaba žičnih povezav. Če to ni mogoče (npr. senzor za detekcijo odprtih polken), pa uporabimo standardne, varne načine brezžične komunikacije (Bluetooth). Izogibamo se protokolom, ki že imajo znane ranljivosti (pazimo tudi, katero verzijo protokola uporabimo).

2.3.7 Zlonamerne aktivnosti oz. zlorabe

V tej skupini imamo več groženj:

- vdor v sistem in kraja poverilnic, neavtoriziran dostop, uporaba ali sprememba (nameščanje škodljive programske opreme) sistema;
- napad DoS ali DDoS:
  - napad DoS (ang. Denial of Service, sl. zavrnitev storitve) je vrsta napada, pri katerem napadalec tarči pošlje veliko število paketov in tako zasiči možne povezave (običajno paket SYN pri protokolu TCP), kar privede do nedosegljivosti naprave [24]. Podoben je napad DDoS (ang. Distributed Denial of Service; sl. porazdeljena zavrnitev storitve), pri katerem sodeluje več računalnikov (zombiji), ki jih nadzoruje napadalec [24];
  - lahko se zgodi, da ena od naprav v sistemu povzroči tak napad;

- večina kupljenih naprav (usmerjevalniki, pametni hladilniki ...) so že pravi računalniki z nameščeno distribucijo Linuxa, kar omogoča napadalcem razvoj in nalaganje zlonamerne programske opreme;

- naprave, kot so pametne televizije, lahko pridejo že z nameščeno zlonamerno programsko opremo (zadnja vrata, vohunsko programje, nezaželene funkcije), kar lahko napadalci izkoristijo;

- z manipulacijo informacij je mogoče prevarati nekatere senzorje in naprave (prepoznava obraza ipd.);

- s ponarejanjem zapisov lahko napadalci po krivem obtožijo lastnika sistema (če sistem vzdržuje podjetje) zlorab ali izsiljujejo uporabnike sistema;

- če ima nekdo vpogled v naš sistem (vzdrževalec), lahko ta zlorabi naše osebne podatke, pridobljene iz senzorjev in naprav, v različne namere (ciljano oglaševanje, razkritje osebnih podatkov, prodaja osebnih podatkov itd);

- oddaljen dostop do pametne hiše – izvedba določenih akcij na daljavo. Napadalec lahko izkoristi nezavarovano funkcijo;

- ciljani napadi – napadalci izberejo žrtev, preučijo tako fizični kot informacijski del sistema.

2.3.8 Pravne grožnje

Pravne grožnje se bolj dotikajo podjetij, ki pametne hiše nameščajo, saj morajo upoštevati standarde in uredbe.

Omeniti velja, da se moramo pri domači uporabi zavedati nekaterih zakonskih omejitev, ki se razlikujejo od države do države. V splošnem moramo biti pozorni na uporabo skupnih medijev, npr. zraka, saj so le določeni kanali dovoljeni za uporabo komunikacije (433 MHz, 868 MHz, 2,4 GHz, 5 GHz –

velja za Evropo) [27]. Pazljivi moramo biti tudi pri uporabi kamer, saj lahko z njimi snemamo le našo posest, ne pa tudi tuje [28].

Poglavje 3

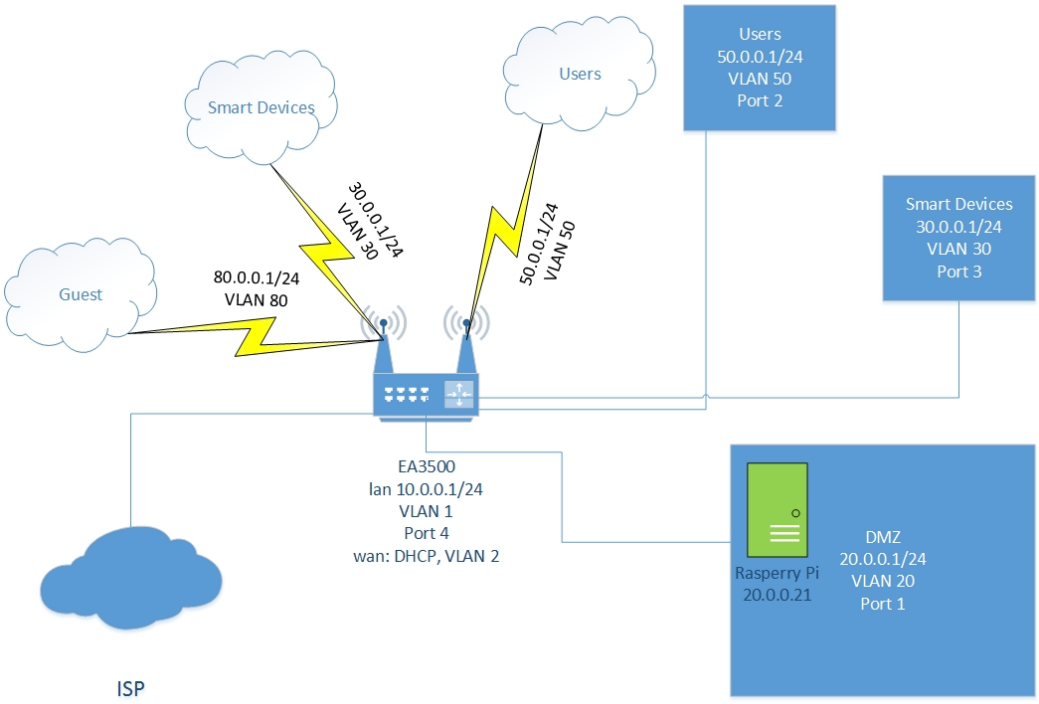
Implementacija

S pridobljenim znanjem in seznanitvijo z dobrimi praksami ter napotki smo realizirali manjši centraliziran sistem s komponentami, ki so nam bile dostopne (predvsem cenovno). Uporabljena strojna in programska oprema:

- usmerjevalnik Cisco Linksys EA3500;
  - OpenWrt (odprtokodna distribucija Linuxa za vgrajene naprave);
- Raspberry Pi;
  - Raspbian (operacijski sistem namenjen namestitvi na računalnik Raspberry Pi);
- Arduino Mega in Arduino Uno;
- moduli za Arduino.

3.1 Omrežje

Na sliki 3.1 je prikazan načrt omrežja. Celotno omrežje sestavlja pet ločenih omrežij, vsako v svojem VLAN-u (ang. Virtual LAN; sl. navidezno lokalno omrežje) in s svojim naslovnim prostorom IP.



Slika 3.1: Shema omrežja.

Omrežje dmz

DMZ (ang. Demilitarized Zone), je fizično ali logično omrežje, na katero so povezani strežniki, ki so javno dostopni. V implementaciji bo dostop do omrežja `dmz` mogoč le uporabnikom, ki so v uporabniškem omrežju. Dostop omogočimo tudi pametnim napravam, katerih naslove MAC (ang. Media Access Control; sl. nadzor dostopa do medija) vpišemo v požarni zid. Zaradi oddaljenega dostopa moramo zagotoviti dostop tudi uporabnikom, ki so na usmerjevalnik povezani prek povezave VPN .

Uporabniško omrežje (user)

Uporabniki pametne hiše imajo dostop do žičnega in brezžičnega omrežja. Pogoj za dostop do brezžičnega omrežja je, da strežnik RADIUS (ta se nahaja v omrežju `dmz`) avtenticira uporabnike.

Omrežje za pametne naprave (smdev)

Pametnim napravam dovolimo dostop le do omrežja `dmz`, do specifične naprave, ki podatke pošilja oz. jih sprejema. Če naprava nujno ne potrebuje internetne povezave, ji le-te v požarnem zidu ne omogočimo. Za omrežje ustvarimo tako žični kot brezžični dostop. Uporabniki iz drugega omrežja lahko do naprav dostopajo samo, če te imajo uporabniški vmesnik. V nasprotnem primeru ni smiselno in niti varno, da lahko uporabniki dostopajo do tega omrežja.

Gostiteljsko omrežje (guest)

V domovih imamo lahko tudi obiskovalce, ki niso uporabniki pametne hiše, zato tem tudi omogočimo dostop do interneta. To omrežje ima omogočena le protokola HTTP in HTTPS (ang. Hypertext Transfer Protocol Secure; sl. varna verzija HTTP), uporabniki so med seboj ločeni, dostop je možen z geslom, nastavimo QoS, da omejimo hitrost od in do uporabnika (nočemo, da nam zasiči omrežje in da nam ali našim napravam internet slabo deluje).

Omrežje lan

Omrežje za dostop do usmerjevalnika je nujo potrebno. Dostop je možen le z žično povezavo. Tako zagotovimo, da nihče ne more dostopati do usmerjevalnika. Dovoljeni so vsi protokoli, edini omrežji, ki se med seboj "vidita", sta `lan` in `dmz`.

Tabela prehodov med omrežji

Tabelo 3.1 beremo kot: "Iz omrežja `lan` lahko dostopamo do omrežja `dmz`."

	lan	dmz	user	smdev	guest
lan	X	✓	X	X	X
dmz	X	X	X	✓	X
user	X	✓	X	X (odvisno od naprave)	X
smdev	X	✓	X	X	X
guest	X	X	X	X	X

Tabela 3.1: Dostop do drugih omrežij.

3.1.1 OpenWrt

OpenWrt je odprtokodna distribucija Linuxa za vgrajene naprave in usmerjevalnike. S konfiguracijskimi datotekami in ogromnim naborom možnosti je mogoče usmerjevalnik nastaviti na poljuben način delovanja. Gre za odprtokodni projekt, pod licenco GPL, z odlično dokumentacijo in skupnostjo, ki projekt širi in ustvarja [17]. To torej ni programska oprema, ki bi jo prenesli in bi že delovala po naših pričakovanjih, ampak je treba vse stvari še nastaviti.

Za usmerjevalnik je bil izbran Cisco Linksys EA3500. Naprava ima naslednje specifikacije:

- procesor Marvell Feroceon 88F6282, ki teče na 800 MHz;
- 64 MB RAM-a;
- 64 MB bliskovnega pomnilnika;
- Bootloader: U-Boot;
- 5 Gb vrat (4 LAN, 1 WAN);
- vhod USB;
- serijski in JTAG vmesnik (napravo je treba odpreti in zaciniti priključek na osnovno ploščo):

- podpora VLAN;
- WLAN 2,4 GHz: b/g/n;
- WLAN 5 GHz: a/n;
- antene WLAN: 2x3:2 na 2,4 GHz in 3x3:3 na 5,0 GHz;
- napajanje: 12V DC, 2,0 A.

Slika 3.2: Usmerjevalnik Cisco Linksys EA3500.

Napravo sem imel v lasti in je v primerjavi z drugimi napravami, ki so trenutno na trgu, še vedno primerljiva in konkurenčna. V primeru nakupa naprave, pa se je treba najprej prepričati, da je nanjo mogoče naložiti OpenWrt, kar lahko storimo na spletni strani OpenWrt [18].

3.1.2 Nameščanje OpenWrt

Na strani kompatibilnih naprav najdemo našo napravo, odpremo stran o napravi, kjer so napisane specifikacije in način namestitve sistema. Na voljo je več načinov namestitve, ki se med napravami lahko razlikujejo. Opisali bomo dva najpogostejša: priporočena je namestitev po prvem načinu, drugega pa uporabimo v primeru, da se kaj zalomi ali ne moremo dostopati do usmerjevalnika.

Nadgraditev naprave

Na računalnik prenesemo ustrezno verzijo OpenWrt (verzije so označene s kodnimi imeni). Povežemo se na usmerjevalnik (če naslova IP ne vemo, odpremo ukazni poziv, vpišemo ukaz `ipconfig`, pod odsekom Ethernet adapter razberemo privzeti prehod (ang. default gateway) in odpremo okno za posodobitev, ponastavitev in nadgraditev programske opreme. Naložimo datoteko (*.bin) in počakamo, da se programska oprema naloži.

Način TFTP

TFTP (ang. Trivial File Transfer Protocol) je preprost protokol, ki odjemalcu omogoča prenos ali naložitev datoteke na strežnik. Specifikacije protokola, ki temlji na protokolu UDP/IP (UDP; ang. User Datagram Protocol; sl. nepovezovalni protokol za prenašanje paketov), najdemo v RFC 1350[8].

Usmerjevalnik je treba najprej ponastaviti na privzete nastavitve. Za namestitev operacijskega sistema na usmerjevalnik potrebujemo datoteko .bin in orodje TFTP Utility [4]. Odpremo ukazni poziv, vpišemo ukaz: `ping 192.168.1.1 -t`. Odpremo TFTP Utility, vpišemo naslov IP 192.168.1.1 in dodamo datoteko v ustrezno polje. Mrežni kabel povežemo na usmerjevalnik in računalnik ter usmerjevalnik ponovno zaženemo. Ko v ukaznem pozivu vidimo, da se je usmerjevalnik odzval, pritisnemo na gumb "Upgrade"in počakamo, da se namestitev konča.

Prvi dostop do usmerjevalnika s sistemom OpenWrt

Na računalniku nastavimo statičen naslov IP 192.168.1.2, privzeti prehod (ang. Default gateway) je naslov IP usmerjevalnika, tj. 192.168.1.1. Z orodjem Putty se povežemo prek Telneta ali SSH (SSH bi moral že delovati, sicer se povežemo preko Telneta in omogočimo SSH; ang. Secure Shell; sl. protokol za varno upravljanje naprav v omrežju) na naslov IP 192.168.1.1. Privzeto uporabniško ime je root.

Ob prvi prijavi posodobimo in namestimo pakete, ki jih potrebujemo

(Nano je urejevalnik tekstovnih datotek, `shadow-useradd` omogoča dodajanje uporabnikov). Dodamo novega uporabnika, mu določimo ime, dodamo geslo in ustvarimo mapo za domač direktorij.

```
$ opkg update
$ opkg install nano
$ opkg install shadow-useradd
$ useradd dejan
$ passwd dejan
$ mkdir /home
$ mkdir /home/dejan
$ chown dejan /home/dejan
```

V datoteki uporabniku na konec vrstice dodamo lupino, v katero se uporabnik prijavi prek SSH. Namestimo paket, ki omogoča preklop na račun root.

```
$ nano /etc/passwd
dejan:x:1000:1000:dejan:/home/dejan:/bin/ash

$ opkg install sudo
```

Zaženemo ukaz `visudo` in odkomentiramo naslednji vrstici:

```
Defaults targetpw
ALL ALL=(ALL) ALL
```

Zadnji varnosti ukrepi za zaščito dostopa so, da uporabniku root onemogočimo prijavo SSH in omogočimo dostop do usmerjevalnika le iz omrežja `lan`. Najprej v datoteki `/etc/passwd` spremenimo del po zadnjem dvopičju v `/bin/false`. Datoteko `/etc/config/dropbear` spremenimo na spodaj navedeni način. Dovolimo prijavo le iz omrežja `lan` (omrežje za usmerjevalnik, dostopno le na enem fizičnem vhodu Ethernet), spremenimo lahko še vrata za SSH, vendar moramo biti še vedno fizično prisotni, da lahko dostopamo do usmerjevalnika (to uredimo v kasnejših nastavitvah požarnega zidu in

omrežja).

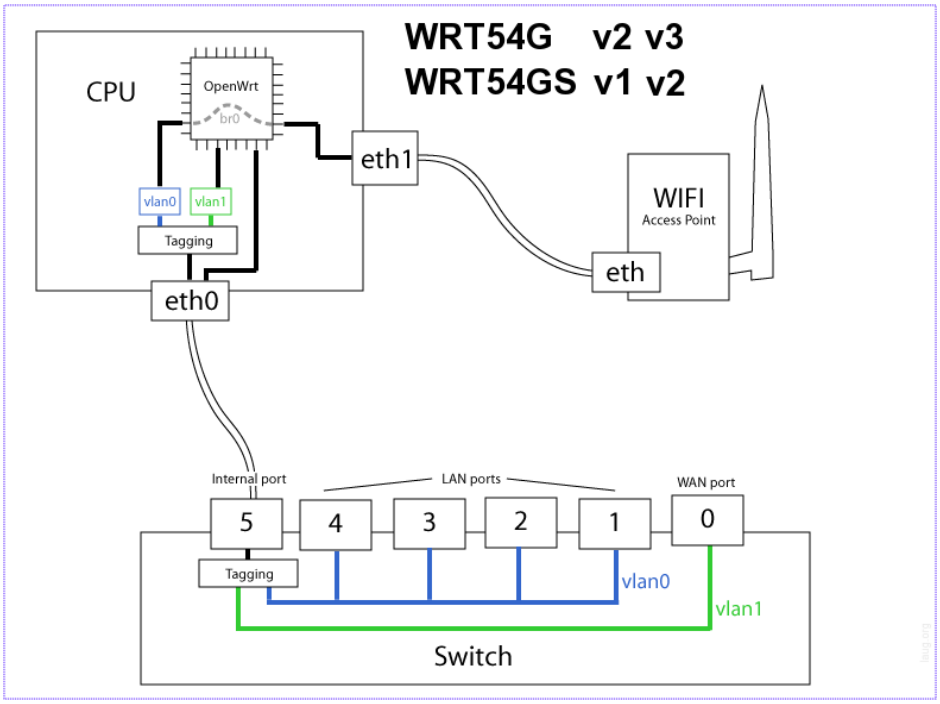
```
root:x:0:0:root:/root:/bin/false

config dropbear
        option PasswordAuth      'on'
        option RootPasswordAuth  'on'
        option Port              '16561'
        option Interface         'lan'
        option RootLogin         'off'
#       option BannerFile        '/etc/banner'
```

Tako smo končali z nastavitvijo varnega dostopa do usmerjevalnika. Če želimo dostopati do usmerjevalnika na drugačen način, lahko to prilagodimo. Če želimo imeti povezave prek spleta, brez VPN-ja, je priporočljiva uporaba javno-privatnih ključev.

Konfiguracija omrežja

Za konfiguracijo omrežja moramo imeti prej predstavljen načrt omrežja z VLAN-i in naslovnim prostorom.

Pozorni moramo biti na dejstvo, da se fizični vhodi ne ujemajo nujno z navideznimi, ki jih uporabimo pri konfiguraciji vrat. Na sliki 3.3 je prikazana zgradba naprave Linksys WRT54G, ki ima podobno razporeditev vrat kot uporabljeni usmerjevalnik (EA3500).



Slika 3.3: Notranja zgradba usmerjevalnika.

Konfiguracijo omrežja napišemo v datoteko `/etc/config/network`. Na začetku datoteke je definirano stikalo, opcija `'enable_vlan'` mora imeti vrednost `1`, da lahko v nadaljevanju ustvarimo navidezne vmesnike in VLAN-e. Od privzete konfiguracije ohranimo še vmesnik `loopback` in vmesnik `wan` (tudi vmesnik `wan6` za IPv6).

Vmesnik `wan` je znotraj usmerjevalnika vezan na vmesnik `eth1`. Da dobi naslov IP, je možnost `'proto'` nastavljena na vrednost `dhcp`. Vhod za priklop internetne povezave je na četrtih vratih (glede na ta ugotovimo, kako so označena ostala vrata), pripisana so še notranja vrata (`6`).

```
config switch
        option name             'switch0'
        option reset            '1'
        option enable_vlan      '1'

config interface 'loopback'
```

```
        option ifname           'lo'
        option proto            'static'
        option ipaddr           '127.0.0.1'
        option netmask          '255.0.0.0'

# wan
config interface 'wan'
        option ifname           'eth1'
        option proto            'dhcp'

config interface 'wan6'
        option ifname           'eth1'
        option proto            'dhcpv6'

config switch_vlan 'eth1_wan'
        option device           'switch0'
        option vlan             '2'
        option ports            '4 6'
```

Kot je razvidno iz načrta omrežja, je treba ustvariti pet omrežij, vsakega s svojim VLAN-om in naslovi IP.

V običajnih primerih je omrežje DMZ izpostavljeno zunanjemu svetu (dosegljivo prek interneta). V našem primeru bomo to prilagodili, saj bodo aplikacija in druge storitve gostovale oz. tekle na strežniku (centralna enota) v omrežju `dmz`, do katerega ne želimo, da imajo uporabniki neposreden dostop prek interneta, temveč le preko povezave VPN. Dostop do omrežja `dmz` je možen le na vmesniku številka nič (na usmerjevalniku so to prva vrata Ethernet), ki se označujejo na petih vratih, (`5t` označuje peta vrata in `t` tagged, zaradi VLAN-ov, ki so speljani čez ta notranja vrata s stikala).

Omrežjem določimo naslovni prostor IP, VLAN, vrata, in če želimo omrežje deliti še na brezžični vmesnik, dodamo tip `'bridge'`. V tabeli 3.2 so prikazana vsa omrežja z nastavitvami.

	switch_vlan	device	vlan	ports	interface	type	ifname	proto	ipaddr	netmask
dmz	eth0_dmz	switch0	20	0 5t	dmz	/	eth0.20	static	20.0.0.1	255.255.255.0
user	eth0_user	switch0	50	1 5t	user	bridge	eth0.50	static	50.0.0.1	255.255.255.0
smdev	eth0_smdev	switch0	30	2 5t	smdev	bridge	eth0.30	static	30.0.0.1	255.255.255.0
lan	eth0_lan	switch0	1	3 5t	lan	/	eth0.1	static	10.0.0.1	255.255.255.0
guest	eth0_guest	switch0	80	5t	guest	bridge	eth0.80	static	80.0.0.1	255.255.255.0
wan	eth1_wan	switch0	2	4 6	wan	/	eth1	dhcp	/	/

Tabela 3.2: Omrežja z vsemi opcijami in parametri zanje.

DHCP konfiguracija

Usmerjevalnik brez konfiguracije strežnika DHCP (ang. Dynamic Host Configuration Protocol; sl. omrežni protokol za dinamično nastavitev gostitelja) ne more dodeljevati naslovov IP napravam, ki so povezana v omrežje. Datoteka `/etc/config/dhcp` zato vsebuje konfiguracijo strežnika DHCP.

OpenWrt vsebuje strežnik dnsmasq DHCP, ki je namenjen vgrajenim napravam. Kljub temu, da je majhen, vsebuje velik nabor funkcij in nastavitev. V omrežju opravlja storitve DNS (ang. Domain Name System; sl. sistem domenskih imen) in DHCP, je torej posrednik za zahteve DNS in DHCP.

Od privzete konfiguracije ohranimo nastavitve za `dnsmasq`, `wan` (naslov IP je pridobljen od ISP-ja) in `odhcpd` (za IPv6) odseke. Novim omrežjem moramo nastaviti interval naslovov, ki jih strežnik DHCP lahko dodeljuje, in čas, ko je naslov IP veljaven (ang. leasetime). Za primer vzamemo omrežje `lan`, vmesnik je `lan` (kar je bilo predhodno nastavljeno v datoteki `/etc/config/network`), naslovi IP za odjemalce se začnejo pri 100 in največji naslov, ki ga DHPC lahko dodeli, je 150. Če imamo naslovni prostor 10.0.0.0/24, so v tem omrežju možni naslovi med 10.0.0.1 in 10.0.0.254, DHCP pa dodeljuje naslove med vključno 10.0.0.100 in 10.0.0.150. Dolžino veljavnosti naslova IP (leasetime) nastavimo na poljubno vrednost, prekratek čas ni priporočljiv, saj naslovi IP odjemalcem pretečejo prehitro in prihaja do novih zahtev in dodeljevanj. V domačih omrežjih, kjer ni veliko uporabnikov, ni težav s predolgimi časi. Tudi za pametne naprave daljši čas ne predstavlja težave, saj naprav ne dodajamo v velikih količinah. Težavo lahko predstavlja prekratek (npr. minutni) leasetime, saj tako po nepotrebnem obremenjujemo naprave v omrežju (nastavimo na urni ali več) in omrežje samo (več paketov oz. zahtev za dodeljevanje novih naslovov IP).

```
config dhcp 'lan'
        option interface        'lan'
        option start            '100'
        option limit            '150'
        option leasetime        '12h'
```

Tabela 3.3 vsebuje vse parametre vključene v konfiguracijsko datoteko `/etc/config/dhcp`.

	wan	**dmz**	**user**	**smdev**	**lan**	**guest**
interface	wan	dmz	user	smdev	lan	guest
start	/	100	100	100	100	100
limit	/	150	150	150	150	150
leasetime	/	12h	12h	12h	12h	12h
ignore	1	/	/	/	/	/

Tabela 3.3: Vrednosti parametrov za DHCP.

Konfiguracija brezžičnega omrežja

V prejšnjem poglavju so bila ustvarjena omrežja in nastavljen strežnik DHCP, do določenih omrežij pa želimo imeti tudi brezžični dostop. Ta omrežja so `user`, `smdev` in `guest`. V konfiguraciji omrežja imajo ta omrežja nastavljen tip na most, kar omogoča deljenje omrežja na vmesnik wifi.

Današnji usmerjevalniki imajo več anten in večina jih deluje na dveh frekvenčnih območjih, in sicer na 2,4 GHz in 5 GHz (ang. dual band). Usmerjevalniki imajo ločene antene za frekvenčni območji, zato sta v nastavitvah vidni kot dve ločeni napravi. Obe napravi moramo imeti nastavljene na `wifi-device`, razlika je v kanalu in `hwmode`, ki pove, ali antena deluje na 2,4 GHz ali 5 GHz. Konfiguraciji naprav ohranimo kot v privzeti verziji, odstranimo le vrstico, ki napravo onemogoči.

Pri brezžičnih omrežjih velikokrat nastane težava s prekrivanjem kanalov.

To rešujemo z namenskimi aplikacijami, ki preverijo vsa prisotna brezžična omrežja, na katerih kanalih operirajo. Glede na pridobljene podatke nastavimo naše omrežje na proste kanale. To je posebej pogost primer v stanovanjskih blokih, kjer ima vsako stanovanje svoj usmerjevalnik z dostopno točko, ki deluje na 2,4 GHz. Druga možna rešitev je prehod na usmerjevalnik dual band (ali ac), ki deluje tudi (oz. zgolj) na 5 GHz. Tak usmerjevalnik ima sicer manjši domet, vendar so tudi težave s prekrivanjem kanalov manjše (enako je z mikrovalovnimi pečicami). Preveriti moramo, da imajo naše naprave (odjemalci) vgrajene mrežne kartice, ki podpirajo to frekvenčno območje, sicer je nakup ali nadgradnja na tako dostopno točko nesmiselna. Preostane le še dogovor s sosedi o uskladitvi omrežja, da se prepreči motnje.

Kot primer nastavitve `wifi-iface` (vmesnika wifi) podamo omrežje `guest`. Opcija `device` pove usmerjevalniku, katero napravo želimo uporabiti (`radio0` smo nastavili na 2,4 GHz, `radio1` pa na 5 GHz). Z opcijo `network` določimo, za katero omrežje nastavljamo povezavo wifi, način delovanja brezžičnega omrežja nastavimo z opcijo `mode`, uporabimo `ap` (dostopna točka), `ssid` (ang. Service Set Identification; sl. indentifikator v brezžičnem omrežju) pa je ime omrežja, kot ga vidimo na naših prenosnih napravah. Pri omrežju `guest` nastavimo `isolate` opcijo na `1`; s tem dosežemo, da se naprave v tem omrežju med seboj "ne vidijo". Še ena nastavitev, ki je le na omrežju `guest`, je največje število asociiranih oz. priključenih naprav naenkrat. Zadnji nastavitvi sta vezani na zaščito omrežja, enkripcijo nastavimo na `psk2` (WPA2-psk) z enkripcijskim protokolom CCMP in kot ključ uporabimo dolg niz, sestavljen iz velikih in malih črk ter številk in znakov.

```
config wifi-iface
        option device           radio0
        option network          guest
        option mode             ap
        option ssid             OW-guest
        option isolate          1
```

```
        option maxassoc          3
        option encryption        psk2
        option key               '#D1pl0msk4N4l0g4'
```

Konfiguracija požarnega zidu

Zadnji del osnovne konfiguracije usmerjevalnika je požarni zid. Za konfiguracijo požarnega zidu potrebujemo tabelo 3.1, po kateri se orientiramo pri pisanju pravil.

Tako kot smo definirali omrežja, sedaj definiramo območja s pripadajočim omrežjem (v eno območje lahko vpišemo več omrežij). Z opcijo `input` nastavimo kaj naj požarni zid privzeto naredi ob prihodu paketa namenjenega za to območje, nasprotno naredi opcija `output`, opcija `forward` se navezuje na pakete, ki so posredovani iz drugega omrežja. Za vsako od teh možnosti imamo 3 možne akcije: `accept`, `reject` in `drop`. Razlika med `reject` in `drop` je v tem, da `reject` zavrže paket in s paketom ICMP (ang. Internet Control Message Protocol; sl. protokol za pošiljanje nadzornih sporočil) pošlje nazaj, da je naslov nedosegljiv (če dovolimo promet ICMP), `drop` pa prispeli paket samo zavrže.

Poglejmo primer za uporabniško omrežje. Nastavimo privzete nastavitve ob prihodu paketa, v pravilih (ang. rules) lahko v nadaljevanju po želji spremenimo, kateri promet omogočimo. Napravam iz drugih omrežij ne dovolimo dostopa do uporabniškega omrežja, obratno pa napravam iz uporabniškega omrežja dovolimo dostop do drugih omrežij. Preusmerjeni promet zavrnemo. Tako konfiguracijo napišemo še za druga omrežja.

```
config zone
        option name              user
        list network             'user'
        option input             REJECT
        option output            ACCEPT
        option forward           REJECT
```

Glede na tabelo št. 3.1 nato napišemo prehode med omrežji. Uporabniško omrežje mora imeti dostop do omrežij `wan` (dostop do interneta), do `dmz` (server z aplikacijo) in do `smdev` (pametne naprave).

```
config forwarding
        option src              user
        option dest             wan

config forwarding
        option src              user
        option dest             dmz

config forwarding
        option src              user
        option dest             smdev
```

Zadnji in najpomembnejši del konfiguracije usmerjevalnika so pravila, s katerimi določimo izjeme, ki se ujamejo pred privzeto vrednostjo. Ob prihodu paketa se preverijo vsa pravila. Če se paket v eno ujame, se to pravilo izvede, sicer se izvede predhodno definirana, privzeta akcija.

Za primer vzamemo nekaj pravil za uporabniško omrežje. Ker ne želimo, da uporabniki dostopajo do omrežja `lan`, napišemo pravilo, ki vse pakete iz območja `user` v območje `lan` (oz. omrežja glede na definicijo območja) zavrže. Enako storimo za omrežji `user-guest`.

```
config rule
        option name             'Deny user -> lan'
        option src              user
        option dest             lan
        option proto            all
        option target           DROP
```

```
config rule
        option name             'Deny user -> guest'
        option src              user
        option dest             guest
        option proto            all
        option target           DROP
```

Napravam iz uporabniškega omrežja dovolimo dostop le do aplikacije, ki gostuje na računalniku v omrežju `dmz`. Vse druge morebitne naprave v omrežju `dmz` pa niso dostopne. Dostop do aplikacije je možen le preko protokola HTTPS.

```
config rule
        option name             'Allow user->dmz https'
        option src              user
        option dest             dmz
        option proto            tcp
        option dest_port        443
        option dest_ip          20.0.0.21
        option target           ACCEPT

config rule
        option name             'Deny user -> dmz'
        option src              user
        option dest             dmz
        option proto            all
        option target           DROP
```

Če želimo v omrežje povezati le naprave, za katere poznamo naslov MAC in ki imajo nenadzorovan dostop do drugih omrežjih, lahko v požarnem zidu za take naprave napišemo pravila. Konkretno so tu mišljene pametne naprave v omrežju `smdev`. Ker te naprave pošiljajo podatke na strežnik, ne želimo, da se katera druga naprava poveže v to omrežje in pošilja ali sprejema podatke

iz strežnika. Primer za tako napravo je lahko pametna sijalka, ki jo povežemo na brezžično omrežje – a lahko pravilo dodamo v požarni zid, moramo poznati njen naslov MAC.

```
config rule
        option src              smdev
        option dest             dmz
        option src_mac          00:e5:e4:43:d0:bf
        option target           ACCEPT
```

QoS

Zagotavljanje kakovosti prenosa nam omogoča, da postavljamo promet v prioritetne razrede. Z namestitvijo paketa `qos-scripts` lahko promet razvrščamo po prioriteti za različna omrežja. V primeru pametne hiše lahko to izkoristimo pri gostiteljskem omrežju. Če imamo slabo internetno povezavo, lahko pride do situacije, ko en uporabnik iz omrežja `guest` zaseda celotno pasovno širino in posledično imajo drugi uporabniki oz. naprave iz drugih omrežij počasnejši dostop do spleta. V ta namen lahko v datoteki `/etc/config/qos` nastavimo pravila za `guest` ali druga omrežja.

Ker ni potrebe po večji natančnosti (imamo nameščen požarni zid), lahko v konfiguracijski datoteki `/etc/config/qos` spremenimo le vmesnik iz `wan` na `guest` in tako dosežemo omejitev prometa do in od nas. Če želimo, lahko na tem mestu za različna omrežja nastavimo različne prioritete za različne protokole.

```
config interface guest
        option classgroup   "Default"
        option enabled      1
        option upload       1024
        option download     256
```

FreeRADIUS

FreeRADIUS vsebuje strežnik RADIUS, programsko opremo pod licenco BSD, ki se uporablja tudi v industriji kot strežnik AAA za Enterprise Wi-Fi in IEEE 802.1X omrežno varnost (eduroam tudi temelji na FreeRADIUSu [25]). V večjih omrežjih je potrebno uporabnike avtenticirati, avtorizirati in spremljati (voditi evidenco prijav ipd.). Uporabnikom se vnaprej določi omrežje, v katero spadajo, in s prijavo v omrežje jim strežnik RADIUS določi VLAN, v katerega spadajo. Prijava v omrežje je izvedena z uporabniškim imenom in geslom.

V primeru pametne hiše se pojavi težava, ko stanovalci želijo dostopati do storitev hiše prek brezžičnega omrežja, saj temu lahko kdorkoli prisluškuje. Čeprav je WPA2 zelo dobro zaščiten, je še vedno možno vdreti v omrežje [22]. Protokol RADIUS to težavo deloma rešuje, saj je za priključitev v omrežje potrebno vpisati uporabniško ime in geslo, kar oteži vdor v uporabniško omrežje. Za omrežje `smdev` to ni potrebno, saj dodajamo naprave z njihovim naslovom MAC (napadalec še vedno ne pozna naslova MAC, ugibanje tega pa zahteva ogromno časa). Prav tako za omrežje `guest` ne potrebujemo RADIUS-a, saj imajo uporabniki tega omrežja omejen dostop do interneta, kar je definirano v požarnem zidu. Zaradi omejitev usmerjevalnika je mogoča realizacija le na brezžičnem omrežju.

Na strežniku, kje je nameščen FreeRADIUS, lahko sledimo navodilom, ki so dostopna na uradni strani. V spodnjem okviru je konfiguracija datoteke `/etc/config/wireless`, v katero vpišemo naslov IP strežnika (`auth_server`) in skrivnost (`auth_secret`).

```
config wifi-iface
        option device           radio1
        option network          user
        option mode             ap
        option ssid             OW-5
        option encryption       psk2
        option auth_server      20.0.0.21
```

```
        option auth_secret          'z3l0m0cn0g3sl0'
        option key                  'D1pl0msk4N4l0g4'
```

OpenVPN

Kot cilj smo si zastavili, da bo aplikacija gostovala na strežniku prisotnemu v hiši, do katerega ni možno dostopati prek interneta. V primeru, da želimo aplikacijo uporabiti tudi, ko nismo doma (nastaviti gretje, hlajenje, svetila ipd.), potrebujemo način, kako na daljavo dostopati do omrežja. Rešitev je navidezno zasebno omrežje. Omogoča nam, da se varno povežemo v domače omrežje; ustvari se varen tunel med trenutnim in domačim omrežjem, s čimer pridobimo domač naslov IP.

Za OpenWrt obstajajo različne možne implementacije VPN-ja. Čeprav je kar nekaj težav z namestitvijo, je priporočljiva je uporaba OpenVPN-ja. Priporočamo uporabo enega izmed vodičev na uradni strani OpenWrt, saj je namestitev daljša, zato bo podrobnejši opis izpuščen.

3.2 Aplikacija

Aplikacija za pametno hišo nam prikaže trenutno stanje pametne hiše, tj. vse podatke s senzorjev in naprav, ter omogoča upravljanje z napravami.

Aplikacija za delovanje potrebuje tri glavne sestavne dele:

Spletni vmesnik
Ta je viden uporabniku, napisan je v HTML-ju (ang. Hyper Text Markup Language; sl. jezik za označevanje nadbesedila), CSS-ju (ang. Cascading Style Sheets; sl. kaskadne stilske podloge) in JavaScriptu. Z uporabo različnih knjižnic JavaScript lahko na pregleden način prikažemo podatke s senzorjev in naprav. Logiko v ozadju (prikaz ustreznih elementov in podatkov) opravlja programski jezik PHP.

Strežniški vmesnik
Strežniška stran je napisana v jeziku PHP, uporabili smo programsko ogrodje Laravel 5 [16], s katerim je razvoj spletnih aplikacij MVC (ang. Model-View-Controller; sl. model-pogled-nadzornik) hitrejši in enostavnejši. Spletni strežnik obdeluje zahteve, bere oz. piše v podatkovno bazo in zaganja skripte, potrebne za izvedbo akcij na mikrokrmilniku in napravah.

Podatkovna baza
Za podatkovno bazo smo uporabili odprtokodno implementacijo relacijske podatkovne baze MySQL, ki za delo s podatki uporablja jezik SQL [30]. V bazi so shranjeni podatki s senzorjev in naprav ter podatki, potrebni za delovanje aplikacije.

Poleg naštetih komponent sta v programskem jeziku Python napisani dve skripti (ena za branje, druga za pisanje na serijska vrata), ki služita kot vmesnik med mikrokrmilnikom Arduinom in aplikacijo. Z Arduinom komunicirata prek serijske povezave.

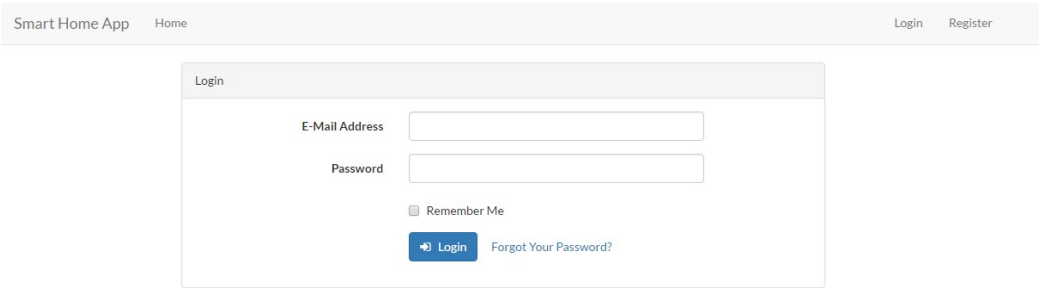
V naslednjih poglavjih so naštete komponente podrobneje opisane, poseben poudarek pa je na implementaciji varnosti pri posameznih korakih.

3.2.1 Čelni del sistema

Za ogrodje spletnega vmesnika smo uporabili že obstoječo predlogo Gentellela Admin, prosto dostopno na repozitoriju GitHub [13]. Napisana je v jeziku HTML, za prilagajanje različnim napravam sta uporabljeni knjižnici Bootstrap 3 in jQuery. Za prikazovanje podatkov (grafi, ikone ipd.) so uporabljene različne knjižnice, ki so pogoste pri razvoju čelnega dela sistema (ang. front-end).

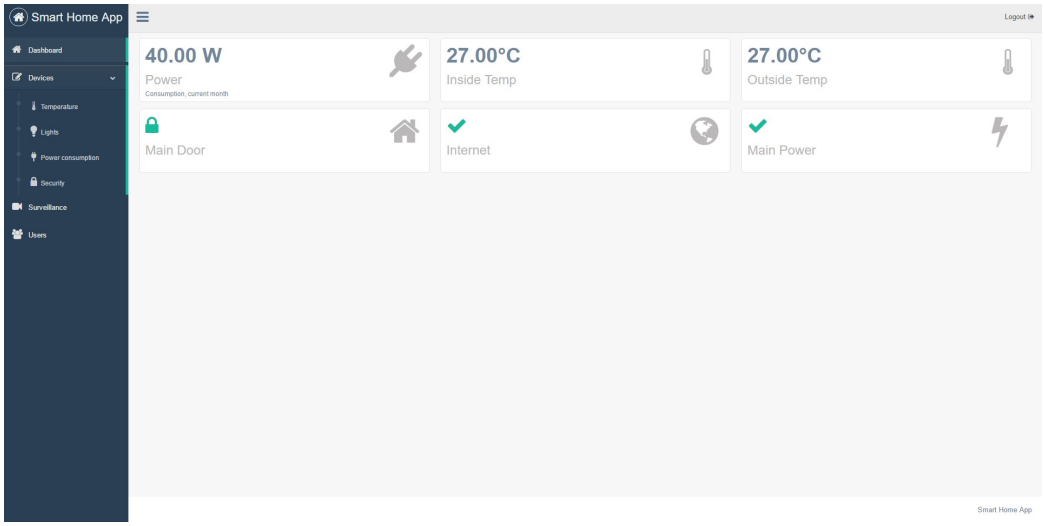
Kot je bilo že omenjeno, se priporoča uporaba programskih rešitev, ki so že napisane in preverjene. Laravel ima veliko skupnost, ki pripomore k razvoju ogrodja, zato lahko že implementirane rešitve smatramo kot preverjene.

Prva taka rešitev je prijava uporabnika. Uporabljena je privzeta predloga in programska koda, znotraj nadzornika je treba spremeniti le določene parametre podatkovne baze (ime tabele in njenih stolpcev).



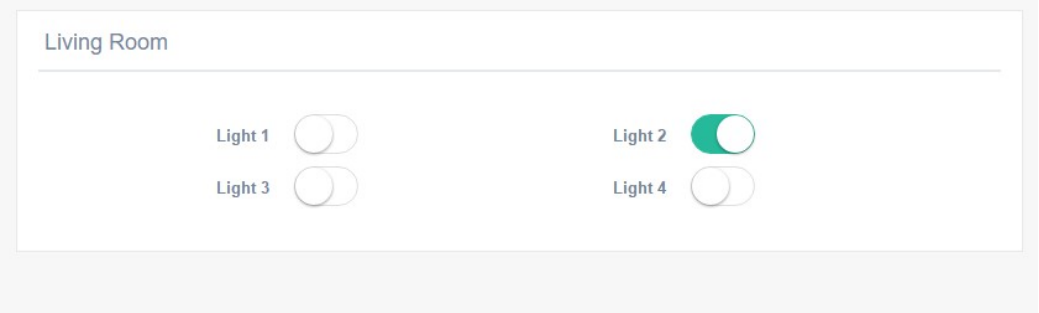
Slika 3.4: Prijavno okno.

Ob uspešni prijavi se uporabniku prikaže nadzorna plošča s podatki o splošnem stanju hiše. Na levi strani je navigacijska vrstica s podrobnejšimi prikazi in funkcijami.



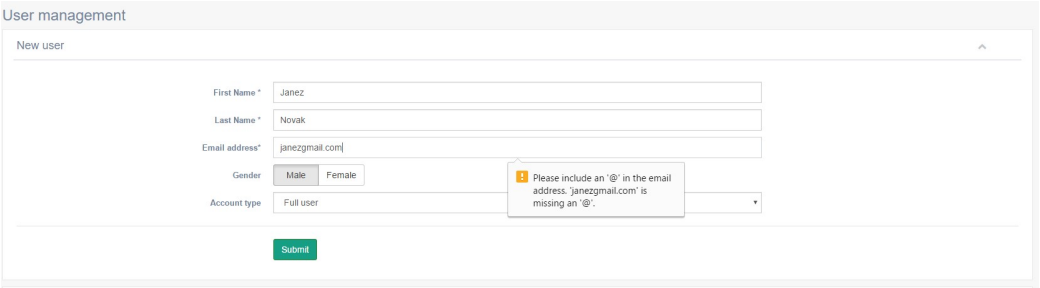
Slika 3.5: Nadzorna plošča.

Kot primer dobre prakse pri implementaciji vmesnikov vzemimo stran za nadzor luči. Uporabnik lahko na prvi pogled vidi, katere luči so prižgane in katere ugasnjene. S preprostim linearnim stikalom (v ozadju je to potrditveno polje, ang. checkbox), dosežemo, da uporabnik ne vnese nobene druge vrednosti. Slab primer bi bil, če bi morali vpisati številko 1 za prižgane luči in 0 za ugasnjene. Uporabnik bi lahko vnesel tudi kakšno črko, kar bi privedlo do napačnega vpisa v podatkovni bazi in napačne interpretacije ukaza. Da ne bi prišlo do napak, bi morali vse preverjati na več mestih. Tako pa s preprostim potrditvenim poljem omejimo uporabnika in si olajšamo implementacijo tako čelnega kot zalednega dela sistema.



Slika 3.6: Vmesnik za upravljanje z lučmi.

Drugi primer dobre prakse so tekstovna polja, ki se najprej preverijo z že vgrajenimi atributi HTML (required, email ...) in programsko kodo JavaScript, po poslani zahtevi pa se preverijo še na strežniku. Tako zagotovimo, da uporabnik ne vnaša napačnih podatkov. Tak princip omogoča ne samo manj napak, ampak tudi stabilno delovanje celotnega sistema. Uporabniki so velikokrat odgovorni za napačne akcije, ki jih s pravilnim dizajnom preprečujemo.



Slika 3.7: Vmesnik za dodajanje novega uporabnika.

Ko govorimo o varnosti odjemalca, veliko težavo predstavlja koda JavaScript. To je programska koda, ki se ob naložitvi strani oz. neki akciji na strani izvede. Poleg HTML-ja in CSS-ja je glavna komponenta modernih spletnih strani z dinamičnimi vsebinami. Programski jezik JavaScript omogoča veliko svobode, ki pri napačni implementaciji lahko izvede nezaželene akcije.

Za primer vzemimo vmesnik za interakcijo z lučmi iz prejšnjega poglavja.

Ob kliku na stikalo se v ozadju sproži koda JavaScript oz. koda, napisana s pomočjo knjižnice jQuery, ki poenostavi programiranje na strani odjemalca. Ta koda izvede zahtevo Ajax in jo pošlje na strežnik. Če se pri implementaciji kode ne upoštevajo vsi možni pogoji izvedbe ali pride do ciklanja, se po nepotrebnem pošiljajo zahteve na strežnik, kar lahko privede do napada DoS na sistem. Zato moramo kodo najprej pravilno implementirati, preveriti vse pogoje izvedbe in testirati s konzolnim izpisom ter na koncu napisati del kode, ki pošilja podatke. Prejete podatke strežnik preveri in izvede ustrezne akcije.

Knjižnice, uporabljene pri implementaciji čelnega dela sistema, so pogosto uporabljene tudi pri izdelovanju spletnih vmesnikov. Skrb za slabo implementacijo je običajno odveč, saj na javnem repozitoriju kode GitHub, kjer so te tudi knjižnice naložene, lahko preverimo, ali obstajajo kakšne napake oz. hrošči, zaradi katerih bi lahko imeli težave.

3.2.2 Zaledni del sistema

Programsko ogrodje Laravel omogoča razvoj aplikacij MVC. Ob dostopu do spletne aplikacije se izvede programska koda v nadzorniku, ki podatke dobi iz modela (podatkovne baze) in vrne pogled (stran aplikacije s podatki iz baze). Pri razvoju zalednega dela sistema moramo biti pozorni na koncepte, kot so avtentikacija uporabnikov, prikaz strani glede na vrsto uporabnika, preverjanje uporabniških vnosov in shranjevanje gesel.

Pri implementaciji s tem ogrodjem imamo olajšan del z avtentikacijo uporabnikov. Ob namestitvi ogrodja ta funkcija ni omogočena, vendar jo je možno namestiti z enim ukazom (`php artisan make:auth`). Popravimo lahko izgled vpisa, registracije in pošiljanja novega gesla ter dostop do podatkov v tabeli podatkovne baze (imena tabel, stolpcev). Na strežniški strani ločimo zahteve URL (ang. Uniform Resource Locator; sl. enolični krajevnik vira) na neavtenticirane in avtenticirane uporabnike. Spodaj je podan primer take uporabe.

```
Route::group(['middleware' => ['web']], function ()
{
    $this->auth();
    $this->get('/', function () {
        return view('welcome');
    });
});

Route::group(['middleware' => ['web', 'auth']],
function ()
{
    $this->get('/', 'DashboardController@index');
});
```

V ogrodju Laravel lahko predloge ustvarjamo s pogonom Blade, datoteke se ločijo po končnici ".blade.php". Pri ustvarjanju pogleda (ang. veiw) iz nadzornika pripnemo še podatke, ki jih lahko prikažemo v predlogi Blade. Tako imamo možnost različnim vrstam uporabnikov prikazati ali ne določene dele aplikacije V spodnjem izseku iz splošnega pogleda je prikazan način uporabe.

```
@if(Auth::getUser()->isAdmin())
  <li>
    <a href="{{ url('/users') }}">
          <i class="fa fa-users"></i>
          Users
        </a>
  </li>
@endif
```

V prejšnjem poglavju smo omenili, da je treba preveriti uporabniške vnose. Za ime in starost določimo pravila, ki jih pred vpisom v podatkovno

bazo preverimo Če pravila niso izpolnjena, se uporabniku prikaže sporočilo o napačno vnesenih podatkih. Primer je podan v spodnjem okviru.

```
private $rules = [
  'name' => 'required|letters',
  'age' => 'required|numbers',
];

$validator = Validator::make($request->all(),
 $this->rules);

if ($validator->fails())
  return back()->withErrors($validator)
    ->withInput();
```

Gesla za dostop do aplikacije so shranjena v podatkovni bazi. Dostop do njih ima skrbnik sistema (dostop do baze in posledično tudi do podatkov v njej). Gesla je treba je zaščititi na tak način, da jih ni mogoče neposredno uporabiti. Rešitev so zgoščevalne funkcije. Ko uporabnik napiše novo geslo, se to prenese na strežnik, kjer se mu doda t.i. sol (običajno je to id uporabnika). Vse skupaj se posreduje zgoščevalni funkciji, s katero dobimo zgoščeno vrednost, ki se vpiše v podatkovno bazo. Ob naslednjem vpisu se postopek ponovi, vendar se primerjata dobljena vrednost (poslano geslo in sol) in vrednost, vpisana v podatkovni bazi.

3.2.3 Podatkovna baza

Podatkovna baza MySQL je nameščena na strežniku, do nje imajo dostop aplikacija, dve namenski skripti (o teh več kasneje) in strežnik RADIUS. Podatkovni bazi sta dve, ena za aplikacijo, druga za RADIUS.

Podatkovna baza se ob namestitvi aplikacije ustvari z ukazom, ki požene migracije. Določene tabele v bazi se zapolnijo s podatki, ki so napisani v datotekah Seeder – prvi vpisani podatki so običajno za testiranje in za ustvarjanje

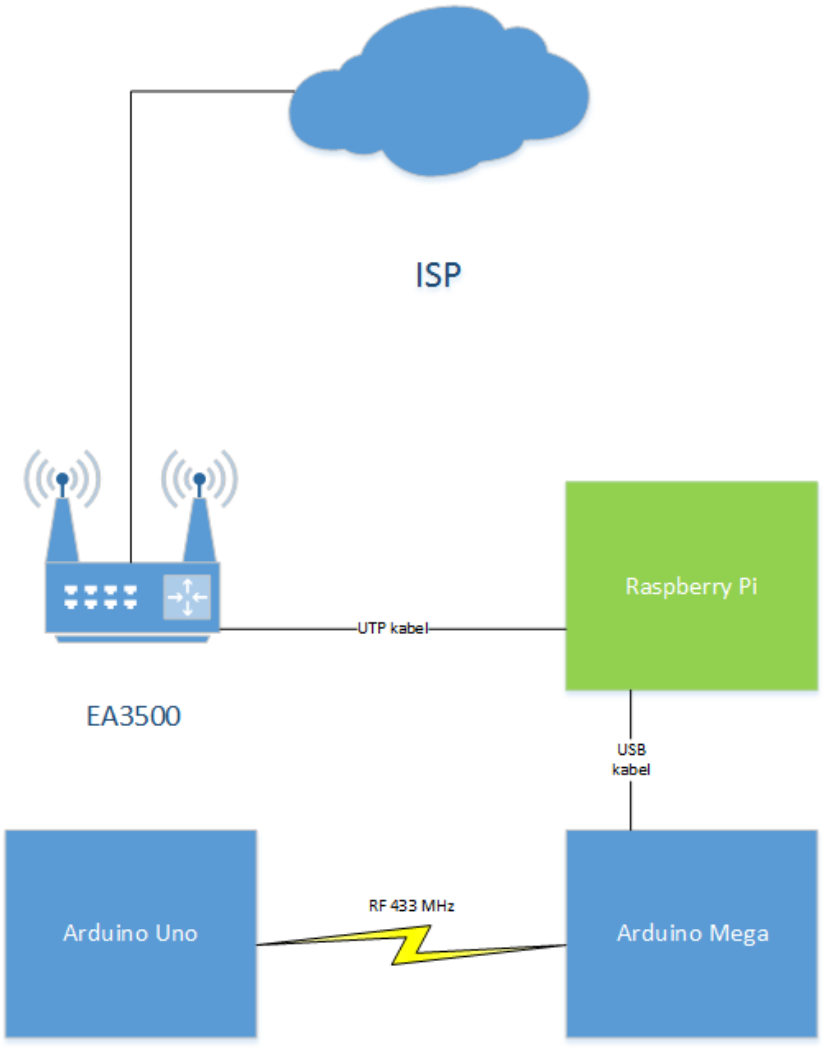
začetnih uporabnikov (administrator). Ukaza (veljata za razvojno orodje Laravel), ki podatkovno bazo ustvarita in napolnita, sta prikazana v spodnjem okviru. Izvede se ju v konzoli, odprti v delovnem direktoriju, v katerem razvijamo aplikacijo.

```
php artisan migrate
php artisan db:seed --class=UsersTableSeeder
```

3.3 Raspberry Pi in Arduino

Za stik med aplikacijo in napravami smo uporabili dve najbolj razširjeni platformi za razvoj projektov, Raspberry Pi in Arduino.

Slika 3.8 prikazuje, kako so naprave med seboj povezane. V naslednjih poglavjih sledi opis komponent in komunikacije med njimi.

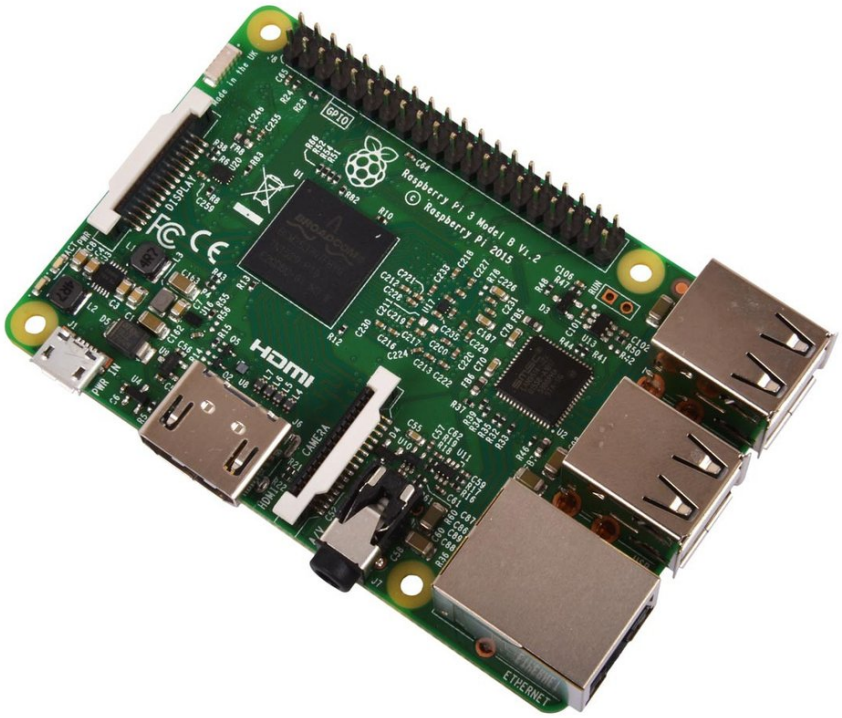


Slika 3.8: Povezave med napravami.

3.3.1 Raspberry Pi

Raspberry Pi je računalnik v velikosti kreditne kartice. Raspberry Pi 3 je zadnja in najzmogljivejša verzija. Specifikacije naprave so naslednje [31]:

- 1,2 GHz 64-bitni 4-jedrni ARMv8 CPE;
- 1 GB RAM;
- 802.11n Wireless LAN;
- Bluetooth 4.1;
- 4 USB vhodi;
- 40 priključkov GPIO;
- vrata HDMI 1.3;
- vrata Ethernet 10/100 Mb/s;
- kombinirani 3,5 mm zvočni priključek in sestavljeni video;
- kartica micro SD;
- VideoCore IV 3D graphics core.



Slika 3.9: Raspberry Pi 3.

Za delovanje naprava potrebuje napajalnik z izhodno napetostjo 5 V in tok vsaj 2 A (2,5 A če so vsi USB-ji zasedeni). Priključimo mrežni kabel, na kartico micro SD naložimo operacijski sistem (naložili smo operacijski sistem Rasbian) in računalnik je pripravljen.

Ob prvem zagonu sistema (postopek za namestitev je opisan na uradnih straneh projekta Rapberry Pi), se prijavimo prek konzolnega dostopa (SSH) in ponovimo postopek menjave uporabniškega imena in gesla za dostop, podobno kot smo to storili na usmerjevalniku. V konfiguraciji požarnega zidu smo za centralno enoto izbrali določen naslov IP, zato moramo v datoteki `/etc/network/interfaces` vnesti statičen naslov IP.

```
auto lo eth0
iface lo inet loopback
iface eth0 inet static
        address 20.0.0.21
        netmask 255.255.255.0
        gateway 20.0.0.1
```

Uporabljena naprava služi kot centralna enota sistema, zato je treba namestiti naslednje komponente, da lahko vzpostavimo delovanje aplikacije in komunikacijo z mikrokontrolerjem:

- strežnik Apache;
- PHP;
- MySQL;
- Composer;
- Python.

Po namestitvi zgornjih komponent ustvarimo podatkovno bazo in naložimo aplikacijo (postopek namestitve in integracije aplikacije se razlikuje od ogrodja do ogrodja, zato podrobnejši opis ni relevanten) ter preverimo njeno delovanje tako, da iz uporabniškega omrežja dostopamo do naslova IP 20.0.0.21.

Konfiguracija strežnika Apache2

Varen dostop do aplikacije zagotovimo tako, da dostopamo le prek varne povezave SSL (ang. Secure Sockets Layer; sl. protokol za varno komunikacijo). Strežnik Apache lahko nastavimo na tako delovanje s spremembo konfiguracijskih datotek.

Omogočimo modul Apache SSL:

```
sudo a2ensite default-ssl
sudo a2enmod ssl
```

Ustvarimo datoteko `/etc/apache2/ssl` in poženemo naslednji ukaz, ki izdela nov certifikat, veljaven 365 dni, in njegov ključ.

```
sudo openssl req -x509 -nodes -days 365 -newkey
 rsa:2048 -keyout /etc/apache2/ssl/apache.key -out
 /etc/apache2/ssl/apache.crt
```

V datoteki `/etc/apache2/sites-available/default-ssl` pod vrstico `Server Admin email` podamo naslov IP ali domeno z vrati: `ServerName 20.0.0.21:443`.
V isti datoteki najdemo spodnje vrstice in jim spremenimo vrednosti v naslednje:

```
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key
```

Nazadnje omogočimo Virtual Host, ki smo ga urejali, in izvedemo ponovno nalaganje servisa Apache2.

```
sudo a2ensite default
sudo service apache2 reload
```

Ponovno poskusimo dostopati do aplikacije, brskalnik nas bo opozoril, da povezava ni varna, vendar če pogledamo digitalno potrdilo, so v tem zapisane

enake vrednosti. V brskalniku lahko še preverimo, ali je povezava šifrirana, tako da v oknu s podatki o strani pogledamo pod zavihek varnost .

3.3.2 Skripti za serijsko komunikacijo

Skripti potrebujemo kot posrednika med pošiljanjem in sprejemanjem ukazov iz Arduina ali iz aplikacije. Arduino povežemo na Raspberry Pi z USB kablom, serijski vmesnik je običajno naprava `/dev/ttyAMA0`.

Arduino - Raspberry Pi

Ob akciji, prejeti na Arduino (npr. vžig luči s tipko), moramo spremembo zapisati v podatkovno bazo. Skripta je napisana v programskem jeziku Python, s pomočjo knjižnice za serijsko komunikacijo.

Raspberry Pi – Arduino

V prejšnjem poglavju o aplikaciji je bilo omenjeno, da se pri določenih klicih funkcij iz spletnega vmesnika izvedejo funkcije, ki pokličejo skripto Python. Skripti so z argumenti podane informacije, ki se pošljejo na mikrokrmilnik, kjer pride do spremembe stanja določene naprave (npr. vžig luči).

3.3.3 Varnost centralne enote

Dostop do centralne enote je možen le iz omrežja `lan`, odprta so le vrata 22, za povezavo SSH. Če je to pravilno nastavljeno, ni treba dodatno zaščititi centralne enote. V večjih okoljih je obvezna tudi zaščita podatkovne baze. Dostop do baze je realiziran z ustvarjanjem različnih uporabnikov, ki imajo dostop do tabel, in z različnimi možnostmi posega v zapise. Ker je edini možen dostop do centralne enote fizičen (lahko tudi VPN), lahko varnostno politiko (dostop do datotek, baze in naprav) nekoliko omilimo in pustimo privzete vrednosti – če je naprava naša in smo mi tisti, ki z njo upravljamo.

Centralno enoto je treba tudi fizično zaščititi. Ne moremo si privoščiti, da bi uporabnik hiše izklopil katerega od kablov (kabel za napajanje, mrežni

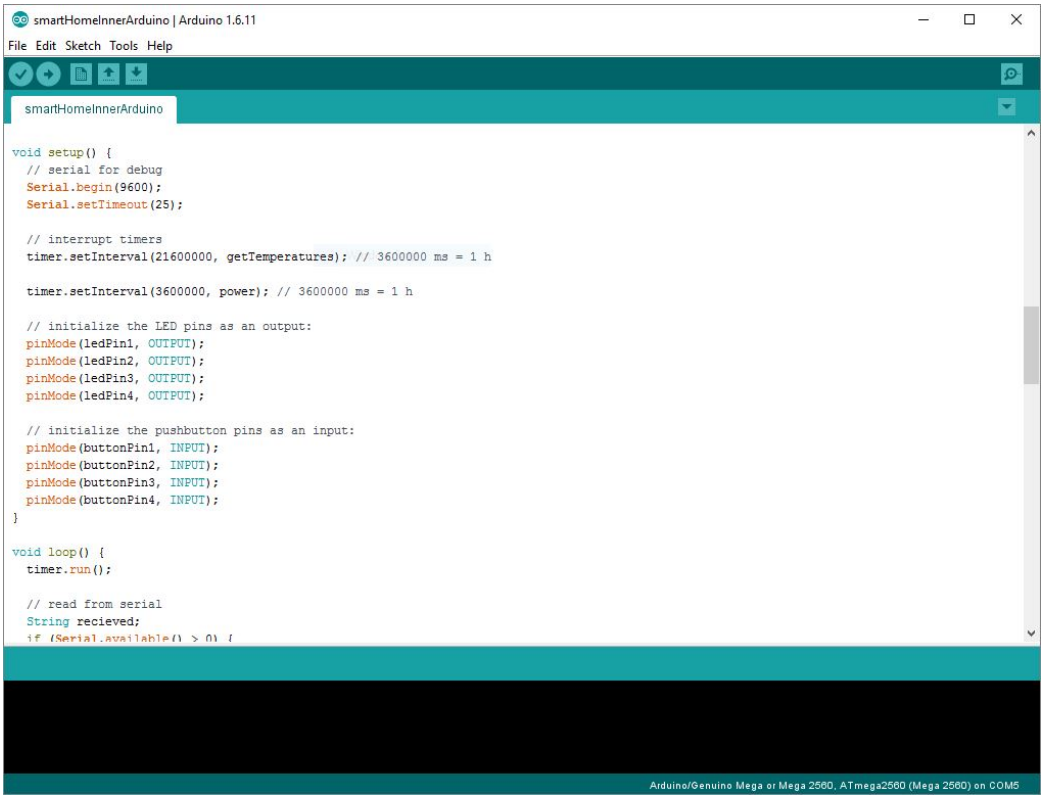
kabel ali USB kabel), saj so ti kritični za delovanje centralne enote.

3.4 Arduino

Arduino je odprtokodna platforma, tako programska kot strojna oprema so prosto dostopne (open source in open hardware) [1]. Platforma je priljubljena predvsem zaradi enostavne uporabe, cenovno dostopnih komponent, odlične dokumentacije in ogromne skupnosti, ki pripomore k razvoju projekta. Zaradi enostavne uporabe, je Arduino primeren tudi za učenje programiranja in elektronike. Obstaja že veliko napisanih primerov programov, zraven katerih so narisane tudi sheme, ki prikazujejo, kako povezati komponente. Na repozitorjih kode najdemo knjižnice, s katerimi si pomagamo pri razvoju projekta. V primeru, da želimo implementirati šifriranje sporočil, uporabimo eno od knjižnic, ki jo je ustvarila oseba s kriptografskimi izkušnjami, in je ne ustvarjamo sami (zahteva po znanju in času).

3.4.1 Arduino Software IDE

Programsko kodo za Arduino pišemo v urejevalniku Arduino, v jeziku Arduino, prilagojenemu programskemu jeziku C oz. C++, ki se prevede v strojni jezik in se izvaja na procesorju. Okolje omogoča enostavno iskanje in uporabo knjižnic, prikaz serijskega okna, nalaganje programa na ploščico Arduino in druge, manj rabljene možnosti. Na spodnji sliki je posnetek zaslona z Arduino IDE.



Slika 3.10: Arduino IDE, z izsekom kode za Arduino Mega.

3.4.2 Arduino Mega in Arduino Uno

V rešitvi smo uporabili 2 razširjena Arduina, model Mega in Uno. Specifikacije obeh so podane v tabeli 3.4 [1].

Na Raspberry Piju, ki služi kot centralna enota, je preko USB kabla priklopljen Arduino Mega. Slednji je bil izbran zaradi velikega števila priključkov (pinov), na katere priklopimo različne module in elektronske komponente (tipke, LED diode ...).

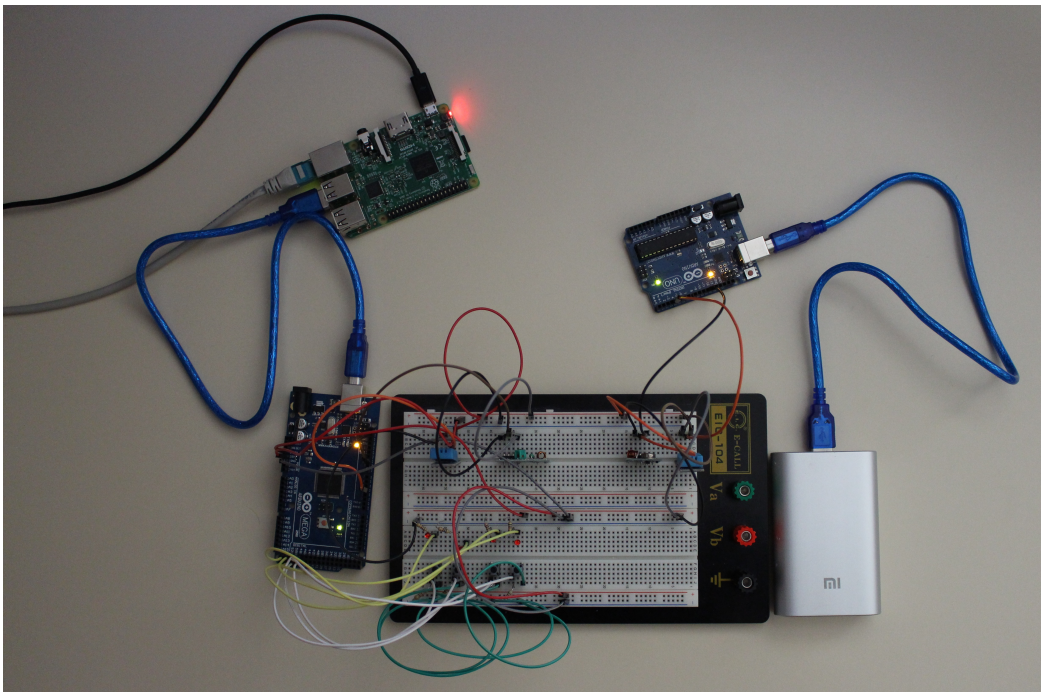
Arduino Mega bere vhod iz tipk, na vhod se odzove in vklopi pripadajočo LED diodo (simulacija luči v hiši). Iz senzorja DH11 bere trenutno temperaturo in vlago. Na Arduino je priključen tudi radijski sprejemnik (deluje na 433 MHz), ki prejema informacije o temperaturi in vlažnosti, poslane z Arduina Uno. Ob prejemu se vsi podatki pošljejo na serijska vrata, na katera

	Mega	Uno
Mikrokontroler	ATmega2560	ATmega328P
Delovna napetost	5 V	5 V
Vhodna napetost	7-12 V	7-12 V
Število digitalnih V/I priključkov	54	14
Število analognih vhodnih priključkov	16	6
Enosmerni tok na V/I priključek	20 mA	20 mA
Enosmerni tok na 3.3 V priključek	50 mA	50 mA
Bliskovni pomnilnik	256 KB (od katerih 8 KB za bootloader	32 KB (od katerih 0.5 KB za bootloader)
Hitrost ure	16 MHz	16 MHz

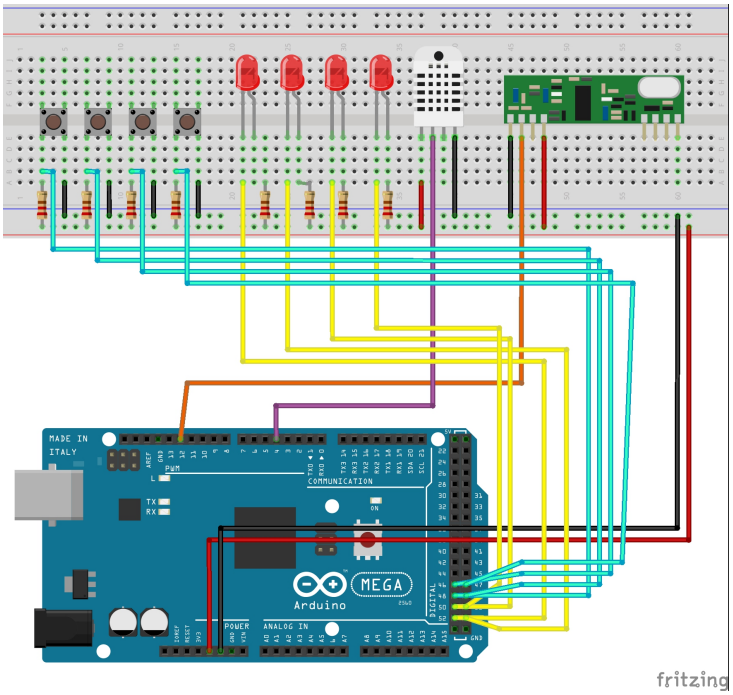
Tabela 3.4: Specifikacije Arduino Mega in Arduino Uno

je priključen Raspberry Pi.

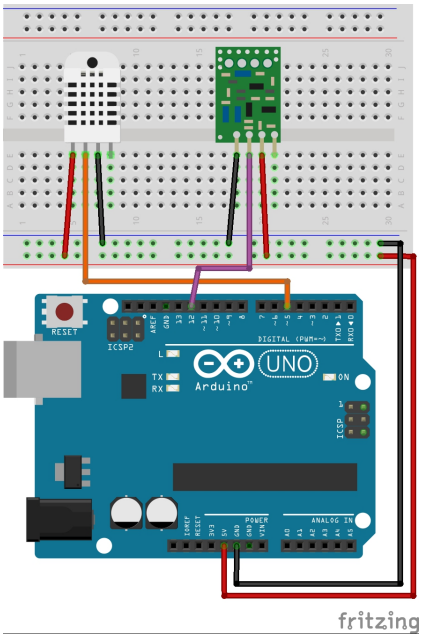
Na sliki 3.11 je razvidno, kako so naprave med seboj povezane. Sledita še shemi (3.12 in 3.13), ustvarjeni v prosto dostopnem programu Fritzing [9].



Slika 3.11: Centralna enota s povezanim mikrokrmilnikom in končno napravo.



Slika 3.12: Arduino Mega, povezava komponent.



Slika 3.13: Arduino Uno, povezava komponent.

3.4.3 Brezžičnen prenos podatkov

Pri delu s platformo Arduino se običajno srečamo z brezžičnim prenosom podatkov. Na Arduinovem forumu je moč zaslediti veliko tem o brezžičnih načinih prenosa podatkov, najbolj izstopajo naslednji moduli: Wi-Fi, ZigBee, Bluetooth in modul RF315/433 MHz. Zanima nas, kdaj je kateri od njih primeren za uporabo, koliko energije potrebuje in kako varen je.

Wi-Fi

Če nimamo centralne enote v neposredni bližini mikrokrmilnika, na katerega so povezane druge naprave, je morda smiselno podatke pošiljati neposredno aplikaciji s specifičnimi naslovi URL, rezerviranimi za mikrokrmilnik. Zavedati se moramo, da Wi-Fi ni primeren za naprave, ki delujejo na baterije, saj moduli potrebujejo veliko energije [5].

Pri nakupu modula Wi-Fi je nujno, da ta podpira protokol WPA2 Personal, ki smo ga omenili v enem od prejšnjih poglavij. Poleg tega je pomembno, da za modul obstaja primerna knjižnica, ki jo posodabljajo.

ZigBee

ZigBee se uporablja za prenos manjših količin podatkov v omrežju. Omrežje ustvarimo z več vozlišči ZigBee, ki lahko med seboj komunicirajo, kar dvema napravama omogoča, da si preko ostalih vozlišč pošiljajo podatke, čeprav se med seboj ne "vidita". ZigBee se uporablja v industriji, v zadnjih letih pa se ga vgrajuje tudi v "pametne"naprave oz. senzorje. Poraba energije je izredno nizka, naprave lahko delujejo tudi več let.

Kljub temu, da ZigBee na višjem nivoju komunikacije zagotavlja visoko varnost (uporaba ključev v omrežju), je znanih nekaj napadov [3].

Uporaba ZigBeeja s platformo Arduino je priporočljiva, držati se moramo dobrih praks, ki jih lahko najdemo na spletu. Tudi poraba energije je primerna – kot smo omenili, naprave delujejo tudi več let.

Bluetooth

Vzpostavitev povezave Bluetooth med modulom za Arduino in drugimi napravami (računalnik, mobilni telefon ...) je enostavna, potrebnih je le nekaj parametrov, napravi se morata popariti in povezava deluje. Ustvari se serijska povezava, pošilja se bajt za bajtom.

Tehnologija Bluetooth ni bila razvita s poudarkom na varnosti [21]. Določene stvari se z leti rešujejo, standard postaja vse bolj prijazen za manjše naprave. Veča se stopnja varnosti in manjša se poraba energije [2].

Modul RF315/433 MHz

Ta modul je eden najcenejših (cene okoli $1) in najbolj uporabljenih modulov v domačih projektih (glede na število navodil dostopnih na spletu). S pomočjo knjižnice VirtualWire je uporaba tega modula povsem enostavna. Knjižnica nudi nekaj varnosti (pošiljanje šifriranega podatka), vendar nekdo s sprejemnikom in to knjižnico lahko prebere tudi vsebino.

Poraba energije je izredno nizka, varnosti pa ne nudi skoraj nikakršne. Iz tega razloga sledi opis zaščite, ki je enostavna in jo je možno uporabiti v vsakem projektu.

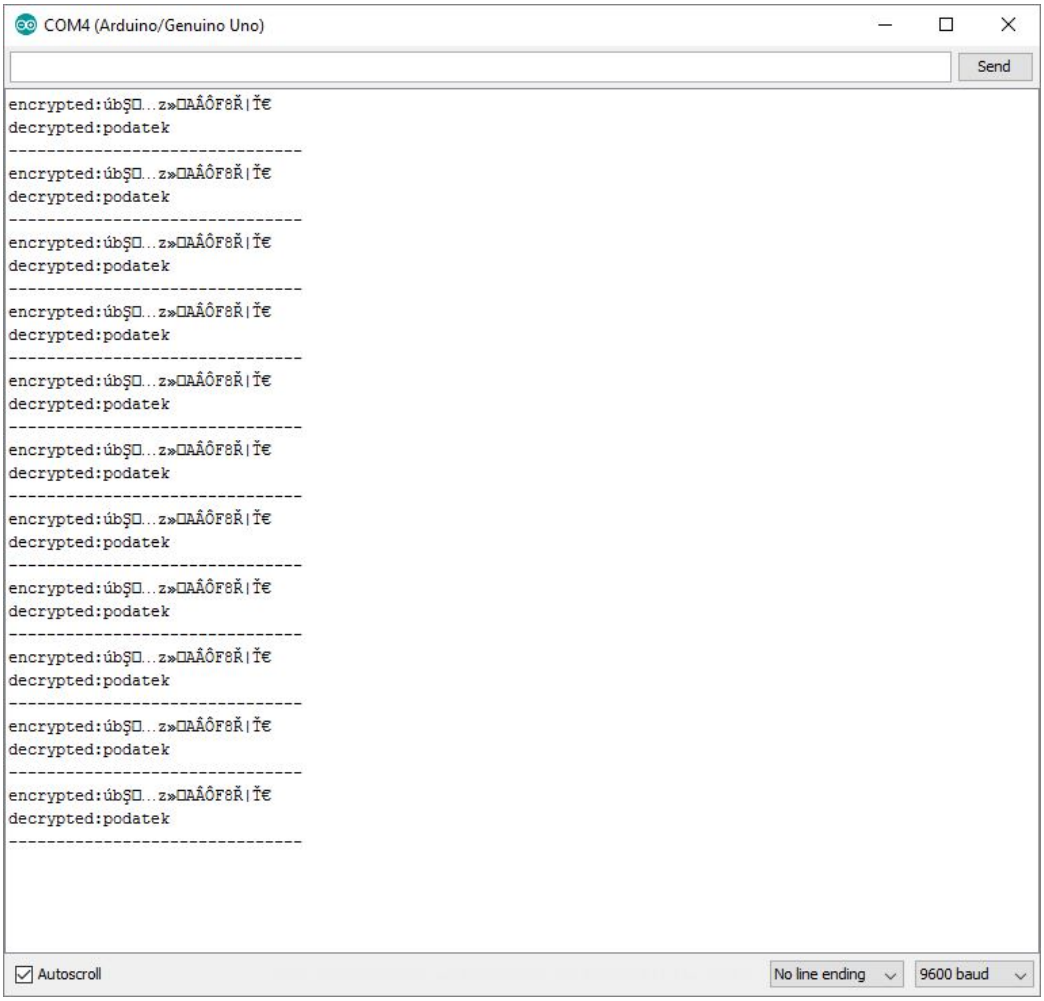
Arduino z modulom RF315/433 MHz

Knjižnica VirtualWire skrbi za integriteto podatkov. To pomeni, da kar pošljemo na oddajniku, pride v enakem vrstnem redu in nespremenjeno na sprejemnik. Težavo povzroča dejstvo, da lahko napadalec to sporočilo prestreže. V ta namen uporabimo knjižnico AESLib, implementacijo simetričnega šifriranja AES na Arduinu. Trenutno so podprti 128-bitni ključi (kar je že dovolj za zaščito sporočil).

Funkciji za šifriranje podamo ključ, inicializacijski vektor, podatek za šifrirat in dolžino podatkov.

```
uint8_t key[] = "6nTp469tP5qh2Rmv";
uint8_t iv[] = "7547763249512385";
char data[] = "podatek";

aes128_cbc_enc(key, iv, data, 16);
Serial.print("encrypted:");
Serial.println(data);
```

Na sliki 3.14 je prikazan izpis serijskega izhoda.



Slika 3.14: Serijski izpis.

Naši podatki so na ta način varni. Ko zaženemo zgornje ukaze v zanki, opazimo veliko težavo. Naša rešitev ni odporna proti napadom s ponavljanjem. Napadalec lahko pridobi šifriran podatek, si ga zapiše in ga pošlje, ko nas ni doma. Izvedla se bo neka akcija, napadalec bo to videl in izkoristil. Zagotoviti moramo, da je poslani podatek vsakič drugače šifriran.

Če sta naši napravi sinhronizirani (uri) ali imata obe sprejemnik in oddajnik, je rešitev za varno pošiljanje in izmenjavo ključev relativno enostavna. Vzemimo primer, da je naš oddajnik daljinski upravljalnik za garažna vrata in vsebuje le oddajnik. Potrebujemo način, da bo poslana vrednost vsakič drugačna, hkrati pa jo bo sprejemnik razumel kot veljavno. Poleg podatka pošljemo še zaporedno številko. Na začetku na obeh napravah je števec enak nič. Oddajnik prvič pošlje podatek in števec se poveča za ena. Sprejemnik najprej prebere prejeti števec, in če je ta večji od njegovega, sprejeti podatek obdela, sicer ne izvede nobene akcije. V realni izvedbi bi potrebovali še zunanji pomnilnik, kjer bi števec zapisovali, saj se ob prekinitvi toka, spremenljivke ponastavijo. Opisan je princip, kako rešiti težave, nadgradnje (menjava ključev in inicalizacijskega vektorja) so v končni izvedbi priporočljive.

Poglavje 4

Sklepne ugotovitve

Število pametnih naprav in celotnih sistemov v domovih se iz leta v leto povečuje [20]. Zanimale so nas grožnje, ki pretijo uporabnikom in sistemu ter kako se jim izognemo.

V prvem delu smo predstavili koncept pametne hiše in kako jo realiziramo. Ugotovili smo, da so sistemi lahko ogroženi in ranljivi. V nadaljevanju smo opisali grožnje, ki so naštete v študiji ENISA [7], za vsako od katerih smo ponudili rešitev, ki bi jo lahko vključili v sam sistem. Grožnje se razlikujejo, težava niso le ranljive tehnologije, tudi uporabniki lahko z nepravilnim ravnanjem ogrozijo delovanje sistema, navsezadnje pa predstavljajo grožnjo tudi naravne nesreče. Pri realizaciji in namestitvi sistema pametne hiše je pomembno, da se teh groženj zavedamo in ravnamo v skladu s primernimi rešitvami.

V tretjem poglavju smo predstavili realizacijo implementacije preprostega sistema pametne hiše. Komponente pametne hiše smo razdelili na več delov: na omrežni del, centralno enoto, mikrokrmilnik in končne naprave oz. module.

Na usmerjevalnik smo uspešno namestili OpenWrt in ga nastavili na željeno delovanje. Največjo težavo pri konfiguraciji usmerjevalnika predstavlja dejstvo, da smo vedno priklopljeni z računalnikom na enega izmed vhodov Ethernet. Težava nastane, če se pri vpisovanju parametrov nekje

zmotimo in ne moremo več dostopati do usmerjevalnika. Temu se je razmeroma težko izogniti; v primeru, da se nam to dogodi (in ponastavitev naprave ne deluje), se lahko na usmerjevalnik priključimo prek serijskega vhoda in dostopamo do konzole (običajno je potrebno spajkati žice na ustrezne priključke).

Sledila je realizacija spletne aplikacije – vmesnika med uporabnikom in sistemom. Pozornost smo namenili dobrim praksam pri razvoju tako čelnega kot zalednega dela sistema. Ker smo privzeli, da bo na centralni enoti nameščena aplikacija, dostop do nje pa omejen, smo omilili varnostne zahteve. Če bi imeli tako aplikacijo nameščeno na oblaku, bi več časa posvetili varnostnim mehanizmom za gradnjo aplikacij.

V zadnjem delu tretjega poglavja smo se posvetili izzivom s platformo Arduino. Žična komunikacija med napravami ne predstavlja težav, te nastopijo, ko želimo brezžično prenašati podatke. Predstavili smo preprost primer, kako varno prenesti podatek, pri tem pa se izogniti morebitnim napadom s ponavljanjem, ki predstavljajo težavo v brezžičnih komunikacijah.

4.1 Nadaljnje delo

Pri izdelavi sistema, smo na uradnih forumih (OpenWrt, Arduino, Raspberry Pi ...) zasledili projekte, ki so bili del (če ne celota) pametne hiše. Sistem, realiziran v diplomski nalogi, lahko služi kot dobra osnova za tovrstne projekte, kot smernice, na kaj moramo biti pozorni in kako naprave namestiti. Zadevo bi lahko tudi nadgradili v spletno stran, kjer bi bile programske in konfiguracijske rešitve predstavljene, razložene in dokumentirane za posamezne tehnologije.

Želja po še varnejšem in robustnejšem sistemu ostaja. Veliko stvari bi lahko nadgradili in izboljšali ter avtomatizirali. Konfiguracijo usmerjevalnika bi lahko avtomatsko generirali glede na nove naprave (dodajanje naslovov MAC v omrežja) v sistemu. Aplikacijo bi napisali kot popolnoma ločena čelni in zaledni del, podatki bi se posredovali neposredno prek API-ja

(ang. Application Programming Interface; sl. aplikacijski programski vmesnik) in ne s skriptami. Ustvarili bi povezavo z oblačno storitvijo, kjer bi delali varnostne kopije, in z napravo NAS. Vpis uporabnikov v sistem, ki ne bi nujno potreboval povezave VPN, bi realizirali z dvostopenjskim vpisom. Na centralno enoto ali usmerjevalnik bi lahko dodali modul za dostop do spleta, za primere prekinitve povezave do ISP-ja. Podobno bi bilo mogoče narediti z električnim napajanjem, imeli bi manjše brezprekinitvene napajalnike, mogoče celo agregate, da bi pametna hiša kljub izpadu elektrike še lahko vedno delovala in uporabnikom zagotavlja varno zatočišče, kar dom navsezadnje tudi je.

Literatura

[1] Arduino. Arduino. Dosegljivo: https://www.arduino.cc/, 2016. [Dostopano 20. 8. 2016].

[2] Inc Bluetooth SIG. Bluetooth low energy. Dosegljivo: https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy, 2016. [Dostopano 2. 9. 2016].

[3] Cisco. ZigBee Wireless Security: A New Age Penetration Tester's Toolkit. Dosegljivo: http://www.ciscopress.com/articles/article.asp?p=1823368&seqNum=4, 2012. [Dostopano 29. 8. 2016].

[4] DD-WRT. TFTP Utility. Dosegljivo: https://www.dd-wrt.com/wiki/index.php/TFTP_flash, 2016. [Dostopano 28. 7. 2016].

[5] Tony DiCola. Low Power WiFi Datalogger. Dosegljivo: https://learn.adafruit.com/low-power-wifi-datalogging/overview, 2014. [Dostopano 29. 8. 2016].

[6] ENISA. Security and Resilience of Smart Home Environments. Dosegljivo: https://www.enisa.europa.eu/publications/security-resilience-good-practices/at_download/fullReport, 2015. [Dostopano 8. 6. 2016].

[7] ENISA. Threat Landscape for Smart Home and Media Convergence. Dosegljivo: https://www.enisa.europa.eu/publications/threat-

landscape-for-smart-home-and-media-convergence/at_download/fullReport, 2015. [Dostopano 8. 6. 2016].

[8] The Internet Engineering Task Force. RFC 1350. Dosegljivo: https://www.ietf.org/rfc/rfc1350.txt, 1992. [Dostopano: 20. 8. 2016].

[9] Fritzing. Fritzing. Dosegljivo: fritzing.org, 2016. [Dostopano 25. 8. 2016].

[10] Google. Google Trends: IoT. Dosegljivo: https://www.google.com/trends/explore#q=iot, 2016. [Dostopano 20.7. 2016].

[11] The PHP Group. PHP. Dosegljivo: http://php.net, 2016. [Dostopano 8. 6. 2016].

[12] Steven Hsie. The Application Trend of Smart Sensing Technology in Home of Building: An Example of a Green and Smart Building for the Seniors Citizens Offered by Farglory Land Development. Dosegljivo: http://www.ausmt.org/index.php/AUSMT/article/view/95/18, 2011. [Dostopano 24. 7. 2016].

[13] MIT Licence. Gentelella Admin. Dosegljivo: https://github.com/puikinsh/gentelella, 2016. [Dostopano 12. 8. 2016].

[14] Icontrol Networks. 2015 State of the Smart Home Report. Dosegljivo: https://www.icontrol.com/wp-content/uploads/2015/06/Smart_Home_Report_2015.pdf, 2015. [Dostopano: 20. 8. 2016].

[15] Christopher Null. The state of IoT standards: Stand by for the big shakeout. 2015. [Dostopano 28. 7. 2016].

[16] Taylor Otwell. Laravel. Dosegljivo: https://laravel.com/, 2016. [Dostopano 12. 8. 2016].

[17] OpenSource Project. OpenWrt. Dosegljivo: https://wiki.openwrt.org/about/start, 2016. [Dostopano 10.7. 2016].

[18] OpenSource Project. OpenWrt Table of Hardware. Dosegljivo: https://wiki.openwrt.org/toh, 2016. [Dostopano 10.7. 2016].

[19] Iztok Saje. Mobilna omrežja v izrednih razmerah. Dosegljivo: https://lms.uni-mb.si/vitel/14delavnica/clanki/iztok_saje.pdf, 2007. [Dostopano 22. 7. 2016].

[20] Johan Svanberg. Smart Homes and Home Automation. Dosegljivo: http://www.berginsight.com/ReportPDF/ProductSheet/bi-sh4-ps.pdf, 2016. [Dostopano 8. 6. 2016].

[21] Vuontisjärvi Tommi Mäkilä, Jukka Taimisto in Miia. Fuzzing Bluetooth Crash-testing bluetooth-enabled devices. Dosegljivo: http://fte.com/docs/Frontline_wp_Fuzzing_Bluetooth_20110919.pdf, 2011. [Dostopano 2. 9. 2016].

[22] Kumkar Vishal, Tiwari Akhil, Tiwari Pawan, Gupta Ashish, and Shrawne Seema. Vulnerabilities of Wireless Security protocols. Dosegljivo: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-1-ISSUE-2-34-38.pdf, 2012. [Dostopano 30. 7. 2016].

[23] Peter Vitez. Kaj je pametna hiša. Dosegljivo: http://luks.fe.uni-lj.si/sl/studij/SUIS/seminarji/peterv/kaj_je_pametna_hisa.htm, 2005. [Dostopano 22. 7. 2016].

[24] Wikipedia. Denial-of-service attack. Dosegljivo: https://en.wikipedia.org/wiki/Denial-of-service_attack, 2016. [Dostopano 23. 7. 2016].

[25] Wikipedia. Eduroam. Dosegljivo: https://en.wikipedia.org/w/index.php?title=Eduroam&oldid=732441195, 2016. [Dostopano: 10. 8. 2016].

[26] Wikipedia. IFTTT. Dosegljivo: https://en.wikipedia.org/wiki/IFTTT, 2016. [Dostopano 23. 7. 2016].

[27] Wikipedia. ISM band. Dosegljivo: https://en.wikipedia.org/wiki/ISM_band, 2016. [Dostopano 23. 7. 2016].

[28] Wikipedia. ISM band. Dosegljivo: https://www.uradni-list.si/1/content?id=50685, 2016. [Dostopano 23. 7. 2016].

[29] Wikipedia. Man-in-the-middle attack. Dosegljivo: https://en.wikipedia.org/wiki/Man-in-the-middle_attack, 2016. [Dostopano 23. 7. 2016].

[30] Wikipedia. MySQL. Dosegljivo: https://sl.wikipedia.org/wiki/MySQL, 2016. [Dostopano 12. 8. 2016].

[31] Wikipedia. Raspberry Pi. Dosegljivo: https://en.wikipedia.org/wiki/Raspberry_Pi, 2016. [Dostopano 20. 8. 2016].

[32] Wikipedia. Unified threat management. Dosegljivo: https://en.wikipedia.org/w/index.php?title=Unified_threat_management&oldid=707546600, 2016. [Dostopano 23. 7. 2016].